A semantic-driven framework for IT support of clinical laboratory standards

Fatima Sabiu Maikore

A thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

Department of Computing
Goldsmiths, University of London

2020

Declaration of Authorship

I, Fatima Sabiu Maikore, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

_____ _____

Signed Date

Acknowledgements

First of all, Alhamdulillah for Allah's mercies and blessings.

I would like to express my deepest gratitude to my supervisor, Dr Larisa Soldatova, for tirelessly and patiently holding my hands every step of the way. You truly are the greatest supervisor of all times.

I also want to thank Daddy and Mama for your love, generous support, and always believing in me. My siblings, thank you for the words of encouragement and help especially my team of expert baby sitters and second mothers to my children.

To Abi, I can finally say I am glad you didn't let me quit, like everything else, I couldn't have done it without you and I am forever grateful to have you in my life. To my children, thank you for putting up with me through it all, you mean the world to me and no words can express my love for you.

Finally, I would like to dedicate this research work to Aunty Jummai. Thank you for being my anchor even when you didn't know that you were. I pray that this research be sadaqatul jariya for you, for every person who benefits from it, may Allah give you the reward and may He grant you the highest station in Jannatul Firdaus.

Abstract

The clinical laboratory plays a critical role in the delivery of care within the healthcare system by providing services that support accurate and timely diagnosis of diseases. The clinical laboratory relies on standard operating procedures (SOP) to provide information and guidance on the laboratory procedures. To ensure an excellent standard of clinical laboratory services, SOPs need to be of high quality, and practitioners need to have easy access to information contained within the SOPs. However, we argue in this thesis that there is a lack of standardization within clinical laboratory SOPs, and machines and human practitioners have difficulties accessing or using the content of SOPs. This thesis proposes a solution to challenges regarding the representation and use of SOPs in clinical laboratories (see Chapter 1). The research work in this thesis is based on the most up-to-date technological, theoretical, and empirical approaches (see Chapter 2). Additionally, external researchers have already utilized the outcome of this research for various purposes (see Chapter 5). In this thesis, we present the SmartSOP framework, a semantic-driven framework, that supports the representation of clinical laboratory procedure concepts in a standardised format for use within software applications. The SmartSOP framework consists of three main components, the Ontology for Clinical Laboratory SOP (OCL-SOP), the translation engine that converts free text SOPs to a standardised format, and a mobile application to provide lab practitioners with easy access to SOPs (see Chapters 3 and 4). We used the design science approach for the execution of this research work.

Contents

List of Figures

List of abbreviations

ADNI	Alzheimers Disease Neuroimaging Initiative
BAO	BioAssay Ontology
BFO	Basic Formal Ontology
ChEBI	Chemical Entities of Biological Interest
CPOE	Computerised Provider Order Entry
CSF	Cerebrospinal Fluid
cTAKES	clinical Text Analysis and Knowledge Extraction System
EXACT	Experimental Actions (ontology)
HBP	Human Brain Project
HHEK	Human Haemoglobin Elisa Kit
IAO	Information Artifact Ontology
MDD	Master Drug Dictionary
MIREOT	Minimum Information to Reference and External Ontology Term
MTERMS	Medical Text Extraction, Reasoning and Mapping System
NCBO	National Center for Biomedical Ontology
NCIT	National Cancer Institute Thesaurus
NDDO	Neurodegenerative Disease Data Ontology
NER	Named Entity Recognition
NHS	National Health Service
NLM	National Library of Medicine
NLP	Natural Language Processing
OBI	Ontology for Biomedical Investigations
OBO	Open Biomedical Ontologies
OCL-SOP	Ontology for Clinical Laboratory Standard Operating Procedures
OWL	Web Ontology Language
PATO	Phenotype And Trait Ontology
PHE	Public Health England
PPMI	Parkinsons Progression Markers Initiative
RDF	Resource Description Framework
SMI	Standards for Microbiology Investigations
SNOMED-CT	Systematized Nomenclature of Medicine Clinical Terms
SOP	Standard Operating Procedure
TTP	Total Testing Process
UMLS	Unified Medical Language System
W3C	World Wide Web Consortium
WHO	World Health Organisation
WIDOCO	Wizard for Documenting Ontologies
YTEX	Yale cTAKES Extensions

Chapter 1

Introduction

1.1 Background

The role of the clinical laboratory is to provide services that are crucial to the effective delivery of care in any healthcare system. The services of the clinical laboratory include providing information about and carrying out tests to enable clinicians to diagnose diseases in patients correctly and in a timely manner. The clinical laboratory testing process, known as the total testing process (TTP), consists of three phases, the pre-analytical phase, analytical, and post-analytical phase [58]. The activities carried out in these phases range from the ordering of tests by physicians to the interpretation of results and subsequent diagnosis and treatment of patients. The quality of the entire process depends on the overall quality of the activities in all the phases. The error rate in the hospital laboratory is one of the critical measures of quality. Although studies have indicated a low prevalence of error for the analytical phase compared to the other phases, there is still room for improvement in the TTP [58, 25]. Due to the large volumes of laboratory tests performed globally, even a low prevalence of errors will translate into significant absolute numbers of occurrences, leading to adverse patient outcome [58]. Another vital part of the TTP worth mentioning is the exchange of laboratory data as part of the post-analytical phase. The

impact of quality laboratory services cannot be fully realised if the right healthcare personnel do not receive accurate data at the right time. Laboratories sometimes collaborate to carry out some complex tests by sharing resources, and the results of all tests need to be sent back to the physician to inform diagnosis. Unfortunately, the exchange of these data are sometimes problematic due to differences in measurement standards, terminologies, reporting formats, and methods of test interpretation between different laboratories and hospitals [79]. Because of the significance of the TTP in the total quality of care delivered to patients, healthcare providers should strive to improve the quality of tests by enabling correct test selection, reducing error rates, and optimum sharing of laboratory data.

One of the attempts made by health organisations to improve the quality of the TTP is to standardise the laboratory practices through the development and implementation of Standard Operating Procedures (SOPs) [79]. For example, in the United Kingdom, healthcare practices are standardised in order to improve quality of care and reduce variations in the treatment of patients [68]. The department of Public Health in the UK (Public Health England) in collaboration with the National Health Service (NHS) have developed SOPs in an attempt to offer guidelines and instructions for clinical laboratory procedures [108]. The NHS requires all its clinical laboratories to adopt the prescribed SOPs to enable standardisation and sharing of best practices. The recommendations provided in these SOPs, if followed correctly, have the potential to improve the quality of test results, thereby ensuring that patients receive appropriate treatment and reducing costs [68, 141]. Laboratories use these SOPs for correct test selection, sample collection and handling, while standardised test terminology and units of traceability to ISO standard 17511 are required to ensure equivalency of measurement results [132]. These SOPs also outline safety guidelines for laboratory scientists while they are carrying out the procedures.

Similarly, the World Health Organization (WHO) has developed strategic frameworks and guidelines for strengthening laboratory services in developing

nations [91, 100, 99]. The WHO has a mandate for disease control and prevention and the clinical laboratory plays a vital role in providing timely information for patient management and disease surveillance [91]. The WHO proposes actions to build the capacity of national laboratories; this includes developing SOPs and subsequent monitoring and evaluation of adherence to the SOPs [91]. To ensure consistency in performing laboratory activities, it is essential to develop and make available standard operating procedures (SOP) for the different laboratories at all levels. Their use should be mandatory by all laboratory staff members every time they perform an activity [99]. Unfortunately, few developing countries have established laboratory quality standards that are affordable and easy to implement and monitor [80]. In cases where these standards exist, they are rarely reviewed and updated [100] and monitoring adherence is challenging.

In addition to the SOPs developed and recommended by national and international health organisations, clinical laboratories within hospitals also create their own individualised SOPs. This is to ensure that the SOPs reflect the laboratorys equipment, measurement standards, and also comply with hospital-specific guidelines for operation. However, this sometimes leads to lack of standardisation in the representation of the SOPs as different laboratories use different terminologies.

The SOPs are presented as free-text documents with both soft and hard (printed) copies in the clinical laboratories. The laboratory scientists are required to be familiar with the content of national and hospital-specific SOPs and adhere to all guidelines provided in the documents.

1.2 Problem statement

The aim of this thesis is to propose solutions to pressing issues regarding the representation and use of SOPs in clinical laboratories. In this section, we identified three main research problems that we will tackle.

- SOPs are essential in ensuring that techniques and processes in the laboratory are correctly written and explained, thereby contributing to the quality of services. The issue of lack of standardisation in the representation of SOPs is one that deserves attention as laboratories often lack well-written SOPs [44]. Brinkman et al. report that the representation of SOPs using non-standardized terminology leads to difficulty in the computational comparison of procedures and also in the reproducibility of the results [28]. Another contributing factor is that hospital-specific deviations from generic SOPs are not always standardised or well documented [80]. Standardisation of SOP representation will improve accuracy and completeness of the procedures while enabling efficient exchange and interpretation of testing procedures and results between different healthcare settings. SOPs need to be represented formally using standardised terminologies to enable interoperability and development of computational systems, particularly intelligent systems, to provide the necessary support to clinical laboratories.

- Another major problem is the inability of machines to read and understand the content of the SOPs. SOPs are inherently ambiguous because they are expressed in natural languages which makes it difficult for accurate exchange of information [125]. This problem makes it difficult to use any automated tool to verify the accuracy and completeness of the SOPs or carry out any automated reasoning. SOPs need to be represented formally using machine-readable language to enable interoperability and development of computational systems to provide the necessary support to clinical laboratories.

- The SOP documents exists as free text either in PDF or MS Word formats, with pages ranging from 12 to more than 50. These documents contain a significant amount of background information about procedures, which the laboratory scientists may not need on a day to day basis while carrying out

test. The laboratory scientists find navigating through such documents time consuming, and the search interferes with the actual testing process [79]. This problem is a classical case of information overload and it leads to lack of adherence to SOP. Currently, laboratories also lack sufficient approaches to monitor adherence to SOPs [80].

1.3 Research aim, questions and objectives

This research proposes an automated solution to fill the gap of knowledge in representation and use of SOPs in clinical laboratories. The main aim is to develop a semantic-driven framework that will provide semi-automatic support for clinical laboratory standards. The main research question we set out to answer in this study is: *How can we standardise the representation of clinical laboratory SOP and support their use in the laboratory?* We identified specific research questions based on the research problems presented in the previous section and formulated a set of objectives to address these questions.

Research Question 1 (RQ1): *How can we formally represent the knowledge within clinical laboratory SOPs to allow for a standardised representation?* SOPs are essential for ensuring that techniques and processes in the clinical laboratory are adequately written and explained, thereby contributing to the quality of services. Standardisation of SOP representation will improve accuracy and completeness of the procedures while enabling efficient exchange and interpretation of testing procedures and results between different healthcare settings. Ontologies are proven to be an efficient standard approach for representing terminological knowledge. There is a need for the knowledge in SOPs to be represented formally using ontology to enable interoperability and development of computational systems. The research objectives for this question are:

- **Research objective 1 (RO1):** *to understand the role of ontology for knowledge representation within biomedical natural language processing*

tools.

- **Research objective 2 (RO2):** *to define an ontology for the formal representation of knowledge in clinical laboratory SOPs.*

Research Question 2 (RQ2): *How to automatically convert SOPs represented in natural language to a machine-readable format while minimising loss of essential information* SOPs are expressed in natural language. However, machines cannot read and understand the content of free text SOPs. There is a need for SOPs to be represented formally using machine-readable language to allow accurate exchange of information, use of automated tools to verify the accuracy and completeness of SOPs, and support automation of laboratory procedures. Translating SOPs into ontology-based knowledge representation, or any other standardised format, requires expert skills, is time-consuming and costly. The research objective for this question is:

- **research objective 3 (RO3):** *to develop a translation engine to convert free-text SOPs into a formal machine-readable representation.*

Research Question 3 (RQ3): *How can we present the clinical laboratory SOP to lab scientists in a way that makes it easy for them to access and use information while monitoring their adherence to the guidelines?* The SOPs exists as large free text documents and as a result, the laboratory scientists face the problem of information overload. They find navigating through the SOP documents time consuming, and searching for information interferes with the actual testing process. This process can be frustrating for the laboratory scientists, and often leads to them neglecting the SOP document. Also, laboratories find it challenging to keep track of and monitor usage of the written SOPs. Consequently, there is a need for a more convenient tool for laboratory scientists to access the information in SOPs. The research objective for this question is:

- **Research objective 4 (RO4):** *to design a mobile application for clinical laboratory scientists to have easy access to SOPs and monitor their adherence to guidelines.*

Objective 5: *to evaluate the usability and effectiveness of the proposed framework*

1.4 Research methodology

We adopted design science as the research methodology for this project. In this section, we will explain why design science is a suitable approach, the design science activities we will focus on, and describe how these activities will address our research objectives.

1.4.1 Design science

March and Smith [82] provide a good basis for design science and define it as an approach to create objects to solve real-world problems. Similarly, Johannesson and Perjons [69] define design science as the scientific study and creation of artefacts as they are developed and used by people with the goal of solving practical problems of general interest. In information technology, design science views an artefact as a construct, model, method or instantiation. Design science projects are considered as research because the outcome is not merely an artefact that can be used in practice but also knowledge about the entire creative process of developing the artefact. The starting point in design science research is the existence of a practical problem that researchers need to solve or improve. In addition to thinking and theorising about the real world, design science researchers aim to model, make, and build artefacts and knowledge about them and how they affect their environment to make new worlds [69]. Hevner et al. [60] have traced the origins of design science to Herbert Simons study of Sciences of the Artificial. Simon puts forth the argument that unlike the natural sciences, which is concerned with the way things are, science of the artificial deals with the way things ought to be [119].

For this research, our main aim is to provide a solution to a practical problem, the representation and use of SOPs in clinical laboratories. We propose

this solution as a framework which is an artefact, that will provide the necessary computational support to the clinical laboratory. We have mentioned that artefacts can be constructs, methods, models, or instantiations. Our proposed framework consists of a construct, a model, and an instantiation. In design science, constructs provide a set of vocabularies which we can use to develop models, which are representations of real-world domains. Instantiations allow us to show how to implement constructs, methods and models as real-world systems. In our proposed framework, the ontology consists of a construct, set of terms from SOPs and a model for representing the clinical laboratory procedures. Our proposed framework also consists of instantiations, a translation engine and a mobile application that demonstrates how the ontology can be embedded in real-life practice. Design science is a suitable approach for this research work as it provides a rich set of activities that allow us to create the framework, knowledge about the creation process, and how this framework will affect the practice of SOP representation and usage in the clinical laboratory. The practical nature of this research project, in general, makes design science research a preferred approach since the goal of design science is utility [60].

[95] argued that while carrying out design science research, it is difficult especially for individual researchers and small teams to produce high-impact. They reasoned that in order to obtain high impact result which involve creating an artefact that will have real-world impact, there is need for extensive collaboration and use of multiple research methods. To mitigate this risk, in this research, we employed several research methods to address our research objectives.

In addition to producing an artefact, we also aim to make additions to the scientific body of knowledge with this research and design science offers us the right approach to create a practical tool as well as new knowledge. There is a difference between routine design and design science research. Routine design employs conventional systems development methods to build effective solutions for organisational problems, while design science contributes to the knowledge

base of foundations and methodologies [60]. Johannesson and Perjons [69] argues that although the activities in design science are similar to those in systems development methods, the latter aims at producing an artefact that addresses the problems of a local practice. Whereas, design science aims to produce new knowledge which is relevant to a more global practice while contributing to the scientific body of knowledge [69].

The position of design science in information sciences domain as a methodology, method, paradigm or approach has been shifting and is much debated in the literature. Woodhill [140] clarifies this position by stating that the literal definitions of design and science can be understood as creating future knowledge which is a concept that encompasses all the positions mentioned. Baskerville [17] stresses that design science is not design neither is it a research strategy nor a research paradigm. However, design science comprises a component of design, uses different research strategies, and can benefit from both positivist and interpretivist paradigms as well as critical realism and critical theory [17]. The different nature of design science to other research strategies and paradigms makes it challenging to make direct comparison and suggest an alternative to design science for our research. We agree with the views of Woodhill and Baskerville and will thus incorporate the other approaches within our adoption of design science. We will use different research strategies during our research activities as well as adopt positivist and interpretivist paradigms. Positivism employs research strategies such as experiments and surveys which provides reliable but shallow knowledge [69]. On the other hand, interpretivism uses strategies such as case studies and action research, which generates more in-depth but less reliable knowledge [69]. It is common to apply both paradigms in design science [69]. We will apply interpretivism in our research during the problem explication and requirements definition activities by using focus group discussion and observations. We will apply positivism during the evaluation activity by using experiments and questionnaires.

In the next section, we provide descriptions of all the approaches we will

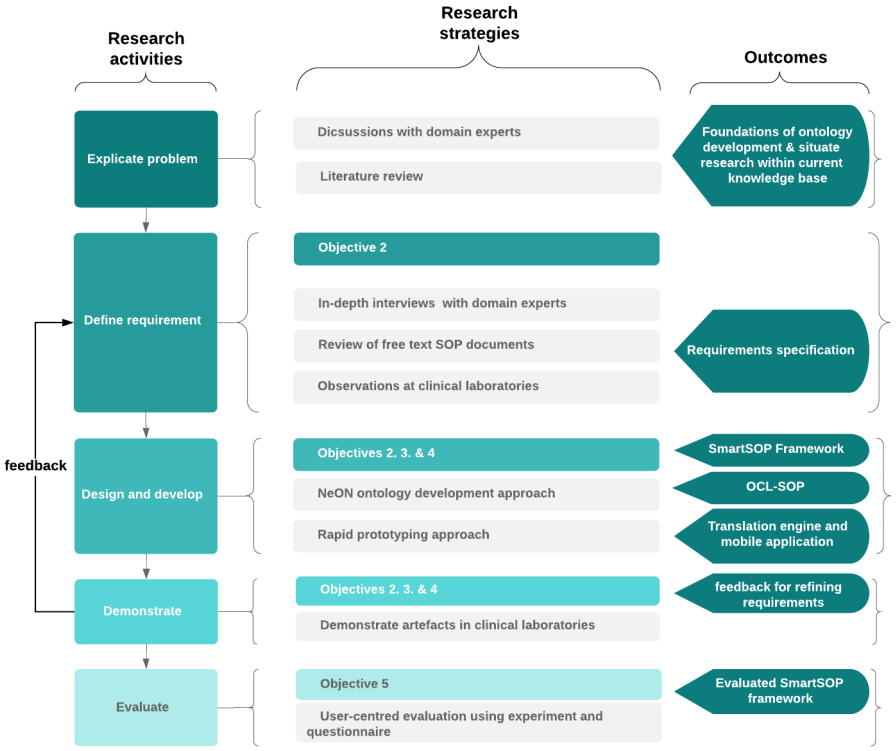

Figure 1.1: Methodological framework of this research

employ in our research activities.

1.4.2 Framework for research strategies and methods

In design science, there are five main activities which are explicate problem, define requirements, design and develop artefact, demonstrate artefact, and finally evaluate artefact. The flexibility of design science allows researchers to focus on a few of these activities while treating the other activities lightly [69]. For this research, we will focus more on defining requirements, and designing and developing the artefact while doing a light problem explication, demonstration, and evaluation of the artefact. The outcomes of the activities we will focus on will be the main contributions of this research. Figure 1.1 shows the different research activities, the combination of research strategies and methods we will use and the outcomes we expect at each stage.

Explicate problem The first activity is the explicate problem, which we will lightly treat because the problems surrounding representation and use of SOPs in the clinical laboratories are already established. We will carry out discussions with a group of domain experts who will constitute of managers and laboratory scientists at clinical laboratories in London (UK) and Zaria (Nigeria). This will allow us to precisely formulate and justify our research problem and show how significant it is in the domain of clinical laboratory practice. For this activity, we chose two locations which have differences in practices to ensure that our proposed artefact will offer a global solution to the research problem. This activity will also address RO1 to give us an understanding of the theoretical foundation for ontology development. We will carry out literature reviews to further understand the research problem and base our proposed solutions on an existing knowledge base. This will make it easier to situate our contribution to existing knowledge. We present the outcome of this activity in earlier sections of this chapter and chapter 2.

Define requirements Our aim for this activity is to create an outline of a solution for the research problem we identified in the explicate problem activity. The solution we propose is an artefact, the proposed framework, and in this activity, we will outline the needs of the clinical laboratory and define how the artefact will solve the research problem. We will conduct in-depth interviews with domain experts from the laboratories in London and Zaria to gather the requirements. We will review free text SOP documents from the collection of UK SOPs, SOPs from the Federal Ministry of Health in Nigeria, and the lab-specific SOPs from the two laboratories. We will also carry out observational studies at the two laboratories by shadowing lab scientists while they are carrying out laboratory testing procedures. The requirements definition activity addresses part of RO2. The outcome of RO2 will be ontology for representing knowledge in clinical laboratory SOP. To develop the ontology, one of the phases of the ontology development approach is defining requirements, which we mapped

to this activity. The outcome of this activity will be a detailed requirements specification document. We present the process of defining the requirements in chapter 3 of this thesis.

Design and develop We will create the different components of the proposed framework, which will address the problem explicated and fulfil the requirements in the design and develop activity. To create the ontology component, we will adopt the NeON approach, which we have already mapped one phase of this approach to the define requirements activity. NeOn is a methodology for ontology development that outlines an approach to reusing existing knowledge sources (both ontological and non ontological) during the design and development of ontologies [130]. For this research, we identified suitable reusable knowledge sources which informed the decision for NeON as the ontology development approach. To develop the translation engine component and the mobile application, we will adopt the rapid prototyping approach. The rapid prototyping approach is suitable for building applications incrementally by adding new functionalities at different levels [84]. This approach will allow us to build the application quickly and more efficiently. We will describe the design and development of the proposed framework in chapters 3 and 4 of this thesis.

Demonstrate For the demonstrate activity, we will use illustrative cases from the clinical laboratory and show how the proposed framework will provide sufficient support for the laboratory procedures. We will demonstrate these cases to the domain experts and get feedback on how to refine the requirements further. Since we are using the rapid prototyping approach to develop components of the framework, we have the flexibility of refining the requirements over several iterations. We will describe the cases we used to demonstrate our framework and prove its feasibility in solving the research problem in chapter 5.

Evaluate The evaluation activity is the last step in our research methodology, where we determined how well the framework fulfilled the requirements

and solved the practical problem defined in our research problem. We will carry out a user-centred evaluation to determine the effectiveness and usability of the proposed framework in representing the clinical laboratory SOPs and providing easy access to the SOPs. We will use an experiment where clinical laboratory scientists will carry out a laboratory procedure while using the mobile application and then gather feedback on the usability using a structured questionnaire. We will present the evaluation process and outcome in chapter 6 of this thesis.

According to [43], design science research methods are very similar to the scientific method. When comparing the design science research activities with the stages of *hypothetico-deductive method*, we can easily draw the following alignments: the explicate problem maps to *ask a question*, define requirements, design and develop maps to *form a hypothesis*, and evaluate artefact maps to *deduce predictions from hypothesis*, and *check predictions* [69]. While the scientific method deals with asking questions and formulating answers as hypothesis, design science research focuses on examining practical problems and creating artefacts to solve those problems. For the context of this research, design science is more suitable as we focused on examining practical problems in the clinical laboratory and creating the SmartSOP framework as a solution to those problems.

1.5 Research contributions

The goal of this research is to provide a solution to the practical problem of SOP representation and usage in clinical laboratories. We set out to create an artefact and drew upon existing knowledge in the field of knowledge representation and its practical use. Our main research contribution is the artefact, **SmartSOP framework** that has the functionality to provide the necessary IT support for working with clinical laboratory procedures. We can decompose our main research contribution into different components as follows:

1. **OCL-SOP ontology**. We built upon an existing ontology for represen-

tation of biomedical experimental actions (EXACT) to create an ontology for clinical laboratory standard operating procedures (OCL-SOP). OCL-SOP provides a novel formal representation of clinical laboratory procedures. Although SOPs exists in free text to provide information about clinical laboratory procedures, their representation is not standardized. We conceptualized the information in free text clinical laboratory SOPs to create a knowledge model. With OCL-SOP, we made a novel contribution to knowledge in the clinical laboratory domain. We described the development and structure of OCL-SOP in chapter 3.

2. **The SOP translator**. We improved an existing translation engine to convert free-text clinical laboratory SOPs into machine-readable formats without losing any vital information. Currently, the SOPs exist in natural language, making it difficult for machines to process information about laboratory process. The SOP translator is a novel contribution to knowledge which uses OCL-SOP as a data model and carries out automatic conversion of free-text SOP into a usable format for smart application. Machine-readable SOPs supports computational comparison of procedures, accurate exchange of information, and automation of procedures in the clinical laboratory. We described the SOP translator engine in chapter 4. In chapter 5, we demonstrated the use of the SOP translator in an external research project to process laboratory protocols for carrying out tests for brain diseases.

3. **SmartSOP mobile application**. We demonstrated the usefulness of machine-readable SOPs by utilizing them in a mobile application (described in chapter 4). Laboratory scientists find it difficult to easily access information about procedures in the free-text SOPs, which leads to problems such as lack of adherence to guidelines that can negatively affect the quality of procedures. With the SmartSOP mobile application, we provide an original contribution that addresses the challenges laboratory scientists

face while trying to access information from SOP. The mobile application reads the machine-readable SOP from the SOP translator and displays the content in a user-friendly format for the lab scientists. The mobile application also allows recording of results from the lab procedures in a machine-readable format.

4. **Evidence of the effectiveness of ontological based tools**. our final contribution is a proof of concept on two levels, the ontological level and the use of ontological data model to solve real-life problems. Through our verification of OCL-SOP, we demonstrated that ontological models allow us to create a specification of shared conceptualization in a domain. We established that OCL-SOP standardizes the representation of concepts in the clinical laboratory domain, thereby facilitating a complete and accurate understanding and sharing of knowledge in that domain by both machines and humans (see chapter 3). We further demonstrated that ontological models could be used as a data model for developing tools that solves real problems through the SOP translator and the SmartSOP mobile application (see chapters 4 and 5). Our evaluation of the SmartSOP framework provides evidence to support these tools provide adequate support for representation and use of SOPs in the clinical laboratory (see chapter 6).

1.6 Thesis outline

The rest of this thesis is organised as follows:

- Chapter 2: This chapter provides the theoretical background of the work we presented in this thesis. The chapter describes issues related to ontology development and the basic formal ontology, which OCL-SOP is based upon. The chapter also reviews the literature on related works and discuss some existing biomedical ontologies, ontology based natural language processing tools, and mobile applications in the clinical laboratory.

- Chapter 3: This chapter describes the development of the OCL-SOP and the structure of the ontology. The chapter explains the NeON methodology we followed to build OCL-SOP, the ontology EXACT that we re-used, and the changes to EXACT. The chapter also describes the verification of OCL-SOP using competency questions.

- Chapter 4: This chapter presents the SmartSOP framework and discuss the three components of the framework. The chapter explains how we OCL-SOP used within the framework. In the chapter we described how we developed the SOP translator and demonstrated how it works. We also described the development of the SmartSOP mobile application and its functionalities.

- Chapter 5: In this chapter, we described past and on-going research collaborations where we used the SmartSOP framework. We described how we aligned the NDDO to the OCL-SOP to enable the use of the SmartSOP framework with protocols for brain disease investigations. We described a second project where we are aligning OCL-SOP to an upper level ontology for representing robot tasks and showed how we can describe the robotic Malaria Microscopy test using the new representation. Finally, we described the ontological components used for developing the Maholo LabDroid and our contribution to the project.

- Chapter 6: This chapter presents user-centered evaluation we carried out to measure the usability of the SmartSOP framework. We evaluated the framework through experiment in several clinical laboratories with lab scientists. In this chapter, we discussed the evaluation approach and presented the results of the frameworks effectiveness in terms of accuracy and completeness of the represented SOP, ease of access to information, and user satisfaction.

- Chapter 7: This chapter provides a conclusion by summarizing the research contributions, limitation of the research, and future work.

1.7 Associated publications

We have presented portions of the work detailed in this thesis in international conferences and journal publications, as follows:

- Chapters 3 and 4: We presented an early version of some of the work from these chapters at the **Joint Ontology Workshops** in Bozen-Bolzano, Italy in 2017. The paper we presented is titled *An Ontology for Clinical Laboratory Standard Operating Procedures* [79]. In this paper, we presented the development and a light weight evaluation of the OCL-SOP and the first version of the mobile application.

- Chapters 4 and 6: We presented the work detailed in sections of this chapter in the paper *A Framework for IT Support of Clinical Laboratory Standards* which is published in **International Journal of Privacy and Health Information Management** [80]. In this paper, we described all the components of the SmartSOP Framework, the OCL-SOP, SOP translator, and mobile application along with an evaluation of the framework.

- Chapter 5: We presented the work in section 5.1 of this chapter in the **International Conference in Discovery Science** in a paper *Neurodegenerative Disease Data Ontology* [73]. In this paper, we described the NDDO and explained how we aligned it to the OCL-SOP.

Chapter 2

Literature Review

In this section, we present background on areas that are relevant for this thesis, which include a theoretical background of knowledge representation using ontologies, the Basic Formal Ontology (BFO), and how BFO can support domain ontology development. We also present related work on existing biomedical ontologies, semantic technologies for processing medical information from free text, and state of the art in laboratory information technology in practice. In sections 2.1, 2.2, and 2.3, we present the outcome of research objective 1, which is to understand the role of ontologies in knowledge representation.

2.1 Ontology

The importance and significance of the semantic web have increased tremendously over the past decade. One of the primary motivations for the development of the semantic web is to make data available on the web for computers to read, interpret, and process in order to generate information and knowledge as well as perform complex tasks. The three essential aspects of the semantic web are the representation of data in a standard format such as the Resource Description Framework (RDF), definition of terminologies using ontologies using languages such as RDF Schema (RDFS) and Web Ontology Language (OWL),

and intelligent software applications to perform complex tasks.

Ontology is one of the fundamental technologies of the semantic web, which allows concepts and their relations in a particular domain to be defined [6]. Staab and Studer define ontology as "a formal description of concepts and relationships that can exist for a community of human and machine agents" [128]. Ontologies facilitate reuse and distribution of knowledge by formally defining a shared conceptualisation. Ontologies also provide the framework that allows a description of human language and real-world notions structurally in a manner that allows machines to read and support the interpretation of such terminologies. Ontology uses a taxonomy and a set of inference rules to describe terminologies [21]. Taxonomy defines classes, which are abstractions of real-world concepts, and their relations or characteristics, which explain how the members of the classes behave and relate to one another. The set of inference rules allow ontology engineers to define the logic of how machines should interpret the meanings of terminologies. This gives power to computers to make inferences and deductions based on existing logic. Ontologies also consist of instantiations of the defined classes. When describing terminologies in a domain for an ontology it is important to define the concepts of the domain as classes, real-world examples of the class as instances, characteristics of the classes as properties, and finally fill in the property values for the instances [113].

2.1.1 Ontology Classification

Ontologies can be classified based on the level of detail about a domain found within the ontology. The main classifications are upper-level ontologies, general ontologies, domain-specific ontologies, and application ontologies. Upper-level ontologies provide very general knowledge without going into details about any domain-specific knowledge [113]. The taxonomy in ontologies arranges terms in a hierarchy of different categories. These categories have various degrees of generalisation with the upper-level ontologies having the most general categories which are reusable across different domains. The main uses of the of

upper-level ontology are to support semantic interoperability of ontologies across domains by providing a common ontological foundation [62]. Some examples of the upper-level ontologies include Basic Formal Ontology (BFO) [122], Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [47], and Suggested Upper Merged Ontology (SUMO) [104].

The second classification of ontologies is the general ontologies, which represents knowledge at an intermediate level of detail without being specific to a particular task [113]. General ontologies also describe generic concepts that are domain-independent such as space and time [34]. One of the most extensive general ontology is the Cyc ontology [76]. The third and possibly the category with the highest number of ontologies is the domain ontology. Domain ontologies define vocabularies which are related to a specific domain, for example, medicine, or business. An excellent example of domain ontology is the Systemized Nomenclature of Medicine Clinical Terms (SNOMED-CT) [37], the most comprehensive clinical ontology. Finally, there are the application ontologies which are to support the functionalities of a specific software application. An example of an application ontology is the MENELAS ontology [24]. It is important to reuse the higher ontologies during the development of domain-specific and application ontologies by refining and specialising concepts from the upper level and general ontologies [34]. This practice will encourage interoperability between different ontologies.

2.1.2 Ontology reuse

The technique of reusing ontologies is becoming increasingly popular due to the realization of the benefits this practice offers. Ontology reuse facilitates knowledge sharing and supports interoperability between systems. We consider ontology reuse here in two ways, reusing existing upper-level ontology for the development of new ontology, and reusing the shared knowledge found in ontologies.

In particular, reusing upper-level ontologies helps the developer to avoid

structural errors and the need for expensive redesign in the future. It also facilitates interoperability with ontologies, which are based on the same upper-level ontologies. Reusing existing ontologies may even be a requirement if the system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies [93]. It is recommended to always look for an existing ontology, which can be reused or extended to address a specific need before deciding to develop a new one. Many ontologies are already available in electronic form, which can be found on the web through libraries of reusable ontologies and imported into ontology-development environment [93]. However, finding relevant ontologies and effectively reusing them is not very easy and is time-consuming. There are some tools available which are designed to support ontology reuse activities. We will describe some selected reuse tools, which we categorized into two, ontology search tools and ontology reuse tools.

The ontology search tools support searching and identifying ontologies that define terms of interest. An example of such tool is the OntoMaton, which was initially built to search for terms in ontologies from the National Center for Biomedical Ontology (NCBO) BioPortal [78]. The BioPortal is a repository of biomedical ontologies (see section 2.4.1). OntoMaton was later extended to search through more than one ontology library with the addition of the Linked Open Vocabularies (LOV), and the EBI Ontology Lookup Service (OLS) [78]. OntoMaton is a Google Spreadsheet add-on which allows its user to search for terms using keywords.

The ontology reuse tools enable the integration of the reused ontology terms into the reusing ontology. In recent years, developers have created a host of ontology development environments (ODEs), which have the capability of importing and thus reusing other ontologies. Among the most popular ontology development tools, there is Protégé [88] which was developed at Stanford University and supports many plugins for importing external ontologies. One of the built-in features of Protégé is the direct import function which enables the user to import an ontology from a specific file or a location on the web. This

functionality enables the user to import an entire ontology or reuse selected terms from the ontology. In order to use selected terms, the ontology developer needs to, first of all, identify the ontology to reuse hence the tools discussed in the previous section will come in handy. There is another tool which can help in reusing selected terms from ontology which is OntoFox [142]. OntoFox is a web-based tool that creates a file and stores only the selected terms the developer wishes to reuse from a particular ontology. This file can then be imported, for example, using Protégé direct import feature. Whether the user is reusing the entire ontology or not, the Protégé import from a web location feature has the advantage of reusing the most up to date term definitions and ontology structure, assuming that the owners have provided the latest version on that specific location.

Protégé also has a range of plugins to support reuse, one of which is the Minimum Information to Reference External Ontology Terms (MIREOT) [57]. MIREOT plugin offers the convenience of searching for terms and directly importing relevant terms within the same ODE. This tool allows the user to either search for terms from a list of available ontologies or load an ontology if it is not on the list. This function is an improvement over OntoFox, where the availability of ontologies is limited. MIREOT is the only tool we came across that is directly on an ODE which makes it easy to use and simplifies adherence to the MIREOT principle. Figure 2.1 shows a screenshot of the MIREOT tool in Protégé.

Features and plugins that support ontology reuse are also available on other ODEs; for example, WebODE [11] allows import and has a functionality for merging ontologies. OntoEdit [120] has plugins which support importing and exporting of ontologies in different standardised formats. Ontology developers need to be aware that these ontology tools are very flexible and will most likely not automatically enforce some of the guidelines and recommendations for ontology reuse. For example, the OBO foundry recommendation for ontology reuse which states that "If an individual term is reused without change to the defini-

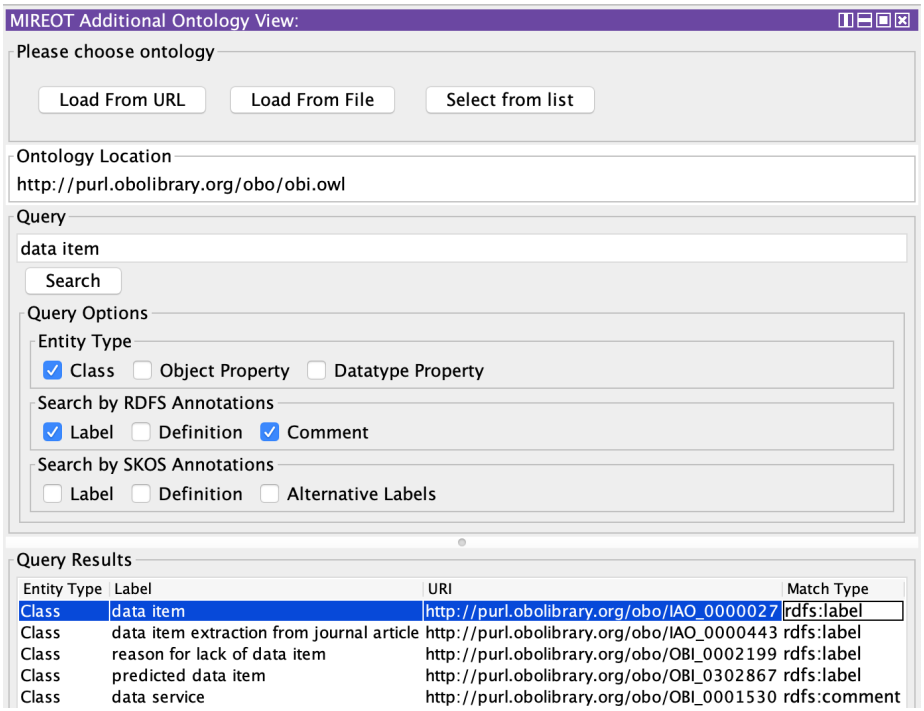

Figure 2.1: Screenshot of MIREOT

tion, the original term IRI (unique identifier) should be used. If the definition of a term (either text or logical) is changed, the original term IRI should not be reused." [121] The first part of this guideline is usually automatically enforced by the tool because it imports all terms with their original IRIs. However, if the developer decides to make changes to the definition of a term, then they need to be aware that the IRI will not automatically change.

It is important also to note that although it is standard practice to make ontologies publicly available to foster reuse, not all ontologies are available to the search tools and some may be out of date.

2.1.3 Web Ontology Language

There are languages recommended by the World Wide Web Consortium (W3C) for ontology development, and each comes with its own set of inference rules. In the beginning of this chapter, we mentioned that ontologies define terminologi-

cal knowledge in a domain by defining classes, instances, properties, and a set of inference rules. W3C developed RDF Schema (RDFS) as an ontology language that allows us to define the underlying semantics of the terminologies [86]. However, RDFS lacks the expressivity to define the complex nature of knowledge found in some domains. The need for a more expressive ontology language gave rise to the W3Cs Web Ontology Language (OWL). OWL takes all the primitives of RDFS and extends it with more expressivity that allows machines to perform useful reasoning tasks on RDF data [86]. The current version of OWL is OWL 2, which was released in 2009. In this section, we will discuss the expressivity of OWL 2 and its different profiles as well as OWL description logic (OWL DL).

OWL 2 expressivity

OWL 2 allows its users to describe entities in ontologies as classes, properties, individuals, and data values using some of the language constructs and axioms we will describe here. The aim of this section is not to give an exhaustive description of all the OWL 2 constructs and axioms but to give a simple description of some common constructs. IRIs uniquely identifies all entities in OWL 2. We will use the RDF turtle syntax for all the examples.

Classes, individuals, datatypes, and literals Classes represent a set of common individuals, which are real-world entities, in an ontology. OWL 2 allows us to describe the relationship between different classes and also define individuals as members of a class. For example, the class of _:Student can be defined as a subclass of _:Person, and an individual _:Sarah can be an instance of and belong to the class of _:Student. This example is expressed in OWL 2 as:

_:Student rdfs:SubClassOf _:Person.

and

_:Sarah rdf:type _:Student.

OWL 2 constructs contain two types of individuals, which are named individuals and anonymous individuals. The difference between the two is that

named individuals are explicitly defined and given IRIs so that they can be used in any ontology while anonymous individuals do not have a global IRI and can only be used within the ontology they are defined in [87]. Datatypes are similar to classes; however, instead of individuals, datatypes represents a set of data values such as strings and numbers [87]. Literals represent actual data values such as a string of characters abc or numbers 123.

Properties In OWL 2, properties are defined to represent relationships between the entities in an ontology. There are three main properties, object properties, data properties, and annotation properties. Object properties show the relationship between two individuals, data properties show the relationship between individuals and literals, and annotation properties provide an annotation for an ontology, axiom, or an IRI [87]. OWL 2 has a set of default properties such as owl:topObjectProperty and owl:topDataProperty but users can also define properties to suit any ontology.

OWL 2 inherits the concept of domains and ranges from RDFS for restricting on how individuals from classes use properties in an ontology. The values of a property are restricting by defining a range while the domain restricts which entities can have the property applied. For example, if we define a new object property as:

_:teaches rdf:type rdf:Property.

Then the property _:teaches can be restricted as such:

_:teaches rdfs:domain _:Lecturer.

_:teaches rdfs:range _:TaughtModule.

Assuming _:Lecturer and _:TaughtModule are classes defined in an ontology, then the above restrictions mean that only an individual from the _:Lecturer class can be related to another individual from the _:TaughtModule class using the _:teaches object property.

OWL 2 supports two kinds of object property expressions which are object property and inverse object property. Inverse object property allow two indi-

viduals who are already connected through an existing object property to be inversely connected to each other. The domains and ranges for a property are reversed for its inverse property. For example, if an object property _:isTaughtBy is defined as the inverse of _:teaches, it is declared in OWL 2 as:

_isTaughtBy owl:inverseOf _:teaches.

And there exists individuals _:ProfPeter as an instance of the class _:Lecturer and _:Statistics as instance of class of _:TaughtModule.

Then this statement

_:ProfPeter _:teaches _:Statistics.

also means

_:Statistics _:isTaughtBy _:ProfPeter.

Class expressions In OWL 2, class expressions or descriptions are formulated from classes and property expressions [87]. RDFS has a mechanism for determining class membership of individual instances using subclass, domain and range. However, OWL 2 allows a more accurate description of conditions that will determine class membership [6]. For instance, membership conditions for a _:Student class are that the student must have a registration number and enrol in at least one module at the university. If any individual at the university meets these conditions, then s/he is considered to be an instance of the _:Student class.

OWL 2 provides a rich set of primitives for use to create class expressions. One way to relate classes that goes beyond the subclass relations in OWL 2 is to use the disjoint union primitive, which does a Boolean combination of the classes. For example, we can define the class of _:Student as a Boolean combination of the class of _:ResearchStudent and class of _:NonResearchStudent and express it as:

:Student owl:disjointUnionOf (:ResearchStudent _:NonResearchStudent).

This means that the individuals in the class _:Student are a combination of all the individuals from _:ResearchStudent and _:NonResearchStudent classes.

While at the same time the disjoint part of the primitive means that the two classes cannot have the same individuals.

OWL 2 formal semantics

In addition to the language constructs, OWL 2 requires a formal semantics to define the precise meaning of the language [6]. Formal semantics allows reasoning about knowledge expressed in a statement. For example, formal semantics of RDFS enables reasoning on class membership given by :x rdf:type :C and :C rdfs:subClassOf :D which allows the inference that :x is an instance of :D. OWL 2 Direct Semantics and OWL 2 RDF-Based Semantics are two ways of assigning meaning to OWL 2 ontologies, with a correspondence theorem providing a link between the two [102]. These two semantics are used by reasoners and other tools, to answer class consistency, subsumption and instance retrieval queries [102].

OWL 2 RDF-Based Semantics is compatible with RDF semantics and uses the conditions for defining the meanings of the RDF language [102]. Since OWL 2 ontologies can be mapped to RDF, the same set of semantics from RDF Semantics apply to the OWL 2 ontologies. RDF documents, which are mapped to OWL 2 ontologies and interpreted with the RDF-Based Semantics, are referred to as OWL 2 Full documents.

OWL 2 Direct Semantics is used to define the meaning of ontology structures directly and is compatible with SROIQ description logic, which is a fragment of first-order logic that has useful computational properties [102]. SROIQ description logic describes the meanings of constructs used for negation and disjoint roles, and for defining properties to be reflexive, irreflexive, antisymmetric, etc. [63, 74]. With some few restrictions, the description logic systems can be used by OWL 2 tools because of the link between their semantics [102]. Ontologies that are interpreted with the OWL 2 Direct Semantics are referred to as OWL 2 DL.

OWL description logic

Description logics are particularly well suited as ontology languages. An ontology language requires a well-defined formal semantics and efficient reasoning support, both of which description logics possess [6, 128]. Description logics are the foundation of W3Cs OWL DL, which is a good choice of language for developing ontologies that will require significant reasoning support while having sufficient expressivity. In the past, there used to be a mismatch between expressive power and the efficiency of reasoning support for description logics, as the more expressive the system is, the less efficient its reasoning support [6, 128]. However, recent research and advancements into description logic systems have significantly reduced this gap [128].

The initial version of OWL consists of three sub-languages, OWL Lite, OWL DL, and OWL Full. These sub languages have varying degrees of expressiveness which was traded off for efficient reasoning support to address the needs of different communities of developers. OWL DL is less expressive than OWL Full but more expressive than OWL Lite. OWL Lite is useful where designers need simple classification hierarchies which makes it easier to provide supporting tools for the sub-language [86]. OWL Full, on the other hand, allows users to use the full expressivity of OWL but with limited reasoning support [86]. OWL DL also adopts the full OWL expressivity but places restrictions on the use of primitives, for example, a resource cannot be a class, property and instance at the same time [6, 86]. Such restrictions are what allows OWL DL to retain some of the reasoning capacity of OWL.

OWL 2 profiles

The disaggregation of OWL into the three sub-languages did not completely address the needs of different communities of developers. For example, some applications use large ontologies which represents complex vocabularies but are concerned with getting computational guarantees over expressiveness [102].

Although these ontologies deal with complicated classifications which needs a precise high-level expressive language, OWL Full will not allow the reasoning support the ontologies require and neither will OWL DL.

In an attempt to address such needs, the W3C created several profiles of OWL 2 that are suitable for different types of ontology development projects. The OWL 2 profiles that exist are OWL 2 EL, OWL 2 QL, and OWL 2 RL. Each of the OWL 2 profiles restrict some of the construct from OWL 2 expressivity as a way to balance the computation needs of the different types of ontologies. All the profiles are more restrictive that OWL DL [102].

OWL 2 EL is an extension of the E L description logic, which can reason in polynomial time on ontologies with a large number of class axioms [6]. This profile is especially useful for large scale ontologies in the health care and life sciences. OWL 2 QL is suitable for developing ontologies with a relatively small set of classes, but a large number of class instances and require efficient query handling [6]. While OWL RL profile enables interactions between description logic and rules and it is the largest syntactic fragment of OWL2 DL that is implementable using rules [6].

Since all the OWL 2 profiles are subsets of OWL 2, then any ontology developed with any of the profiles can be reasoned with either the OWL 2 Direct or RDF-Based Semantics [102]. In order to choose the most appropriate profile for their needs, application developers need to consider the level of expressivity required while giving priority to reasoning on classes as well as the type of data they need to process [102].

2.1.4 Ontology development methodologies

Ontology development is a tedious and time-consuming process, and there is a need for a well-defined methodology. In the last two decades, ontology developers have proposed several methodologies; however, only a few have reached a substantial level of maturity and are widely accepted [67]. Iqbal et al. reported that although there are few methodologies like the METHONTOLOGY, which

has sufficient details, most methodologies reported in the literature lack sufficient details of their techniques and activities [67]. The majority of proposed methodologies arise from the experiences of ontology developers on ontology development projects and adaptation of the software development process. The methodologies have different approaches which includes designing ontologies from scratch or reusing existing ontologies. We have already discussed ontology reuse and the tools available for reuse in section 2.1.2.

Fernndez-Lpez and Gmez-Prez argued that since ontologies are part of software products, they should be developed according to standards for software development, taking into consideration the distinctive characteristics of the ontologies [40]. They identified the IEEE Standard 1074-1995 as the software development process for ontology development methodologies. Their recommendations for each of the processes from the IEEE standard are below:

- Methodologies should recommend software lifecycles for ontology developers to choose from
- Methodologies should implement project management process which includes activities related to project initiation, project monitoring and control, and software quality management
- Methodologies should describe development processes for the ontology which are categorised into pre-development processes (such as feasibility study), development processes (requirement specification, ontology design, and implementation, and post-development process (installation, support, maintenance)
- Methodologies should include integral processes which involves training for maintenance of the ontology

Cyc methodology was formed based on the experiences from the development of the Cyc Knowledge Base, which provides an extensive collection of practical knowledge to provide natural language support to systems [76]. There are three

phases in the Cyc methodology which requires varying degrees of support from automated tools. The first phase requires manual extraction of common sense knowledge from several sources without the use of any tools, the second phase involves the use to machine learning and natural learning tools to support codification of knowledge, and the third phase is carried out mainly by the tools with little intervention from the developer [76, 35]. This approach will work well on large corpus of knowledge which may be difficult to encode without the support of automated tools. One drawback of the Cyc methodology, however, is that it does not recommend life cycle, project management process, and lacks details of the pre and post-development process [35].

Uschold and King developed the first ontology development methodologies based on the experiences from the development of the Enterprise Ontology [135]. The stages in this methodology consists of identifying the purpose of the ontology, building it (includes ontology capturing, coding the knowledge captured, and reusing existing ontologies), and evaluating, and finally documenting the ontology [40, 135]. Although this methodology provides some of the activities from the IEEE standard, it is missing a recommendation for a life cycle, and the pre and post-development processes [40]. The methodology also lacks details of the activities it outlined [67].

Grninger and Fox proposed a methodology based on the experiences from the development of the TOVE project ontology [53]. This project involved modelling business processes and activities [40]. This methodology proposes several steps to develop a logical knowledge model for the ontology using first-order logic [35]. Fernndez-Lpez et al. explained the series of steps for this methodology in [40, 35] as follows:

- The first step if to identify the applications that will possibly use the ontology
- Then determine the scope of the ontology through a set of informal competency questions expressed in natural language

- Then specify the terminology from the answers of the competency question in a formal language
- Then create a set of formal competency questions based on the ontology terminologies
- Then specify the axioms and definitions of the terminologies using a formal language
- Finally, create the conditions for validating the completeness of the ontology

Similar to the methodology of Uschold and King, Grninger and Foxs methodology does not recommend a specific life cycle and lacks some details on the activities and techniques. Likewise, the pre and post-development processes, project management, and design processes are missing [40].

METHONTOLOGY framework is an ontology development methodology that enables the development of ontologies at the knowledge level [41]. METHONTOLOGY supports the development of ontologies from scratch or by reusing (partially or wholly) existing ontologies [35]. So far, METHONTOLOGY is the most matured ontology development methodology available. It has been extensively tested and used in several projects, unlike the other methodologies which are used in a limited number of projects. It proposes a lifecycle that is based on evolving prototypes for identifying the ontology development process, and provides details of the techniques for each of its group of activities [35]. METHONTOLOGY framework proposed a group of activities that synchronise well with the IEEE standards activities. These include the management activities where details of scheduling, control and quality assurance are available. There are also development activities which include specification, conceptualisation, formalisation, implementation. Then there is the group of supportive activities which include knowledge acquisition integration, evaluation, documentation, and configuration management.

NeOn methodology is another ontology development methodology, but it is relatively new compared to the others described above. NeOn methodology framework is a scenario-based methodology that is recommended to speed up the development of ontologies by reusing existing knowledge resources such as ontologies and non-ontological resources [130]. This methodology provides nine scenarios for application, a set of processes and activities for the development process, two ontology life cycle models, and a set of detailed guidelines for different processes and activities [130]. Like METHONTOLOGY, NeOn Methodology aligns well with the IEEE standards activities for software development processes. Although METHONTOLOGY also supports ontology reuse, the entire framework of NeOn is based on reusing existing knowledge sources and gives options for pathways for development. These characteristics of the framework make it more suitable for todays ontology development because the recent boom in ontology development makes available several options of re-usable knowledge sources.

2.2 Ontology as a knowledge representation and theory formation

In recent years, researchers have extensively explored ontology development and its application to various domains such as biomedical, e-commerce, and education. Ontology is a knowledge representation tool that formally specifies and describes concepts in a domain. However, recent works show that ontology also as a method for theory formation [59, 122, 42, 3]. In this section, we will explore the two angles of ontology, both as a tool for knowledge representation and theory formation.

There are two notions of ontology as described in the literature; the first notion which is supported by information scientists describe ontology as software implementation which captures the shared conceptualization in a domain, while the philosophers support the second notion that describes ontology as theories

of entities [122]. It is crucial to bring together these two notions in order to strengthen the representation aspect of ontology and produce robust ontologies. [122] emphasis that using ontological theory to inform modelling decisions is necessary to ensure that the ever-growing complicated terminologies in a domain can remain consistent, which in turn will allow efficient and correct reasoning support. [3] also explains these two ontological notions as the dual reference of ontology where on the one hand it refers to the computational specification for a computer information system, and on the other hand it refers to the theoretical model of real-world domain

2.2.1 Knowledge representation with ontologies

The understanding of ontologies as computer implementation for the representation of knowledge in a domain is the most popular idea. Ontologies allow knowledge representation by providing a set of taxonomy and a set of inference rules. Taxonomy defines classes, objects, and their relationships, found a particular domain. The inference rules allow the ontology to support computational reasoning by allowing deductions to be made based on the taxonomy. We have extensively described what ontologies are, the meaning of the different taxonomy components, OWL, methodologies, and tools for developing ontologies is section 2.1.

One of the most common approach to capturing the terminologies and their meanings from a domain is through consultation with domain experts alongside perusal and review of written documents based on different subject areas of the domain. This approach will allow the domain engineer to create a descriptive semantic representation, i.e. the taxonomy. However, the modelling decisions ought to be informed by a sound ontological theory because of the various benefits identified in the works of [59, 122, 42, 3].

2.2.2 Ontological theory

There are various reasons why ontological theory is relevant, but before we discuss those, we want to emphasise why theory in general matters for software engineering. [29] rationalises that engineering disciplines of which software engineering belongs, need to be based on scientific practices and theory to provide scientific evidence and a justification that their methods work properly. The reason why theory is so critical is that it helps determine and evaluate the concepts that provide the basis for identifying terminology and developing engineering methods [29]. The various arguments defend that theory can improve software engineering by providing robust methodologies and a better understanding of domain-specific knowledge among other uses.

Herre describes formal ontology as an approach for systematic development of axiomatic theories describing forms, modes, and views of being of the world at different levels of abstraction and granularity [59]. In their discussion, an ontology provides a standard approach for communication, allows organisation and representation of knowledge, and contributes to theory formation and modelling of concepts in a domain [59]. The knowledge presented by a system of axioms described in an ontology allows the use of computer-based methods to draw conclusions, create hypothesis, and interpret data.

The entire premise of ontology development is based on an understanding of the concepts in a domain. [3] stipulate that developing this understanding is itself the conceptualisation and theory formation act. This understanding (theory formation) is valuable beyond the ontology engineering as it allows advancement in the application domain through automated reasoning and generation of new knowledge. There is evidence in the philosophy of science literature of the importance and difficulty of theory formation in scientific research [3], which we can remedy through the adoption of ontology development as a tool for theory formation.

Ontologies, when used for theory formation, provide domain experts with a way to model phenomena, and test models through computer simulations and

calculations. The two typical ways of representing theory are either mathematical models or in natural language, which has drawbacks such as difficulty in understanding the theories. In contrast, ontologies provide us with the advantage of using automated simulation tools and graphical and diagram representations of theories which makes it easier to understand [3].

The acceptance of ontological theory is necessary for a robust ontology development approach. In order to fully understand the terminological knowledge in domains, it is necessary to have an understanding of the theories surrounding such knowledge and have a method of adequately capturing such theory. Furthermore, having a very rich and accurate taxonomy from a domain enables theories to be formulated easily, making ontologies a logical theory formulation approach.

2.2.3 Evaluation and validation of ontological theories

The idea of using ontologies as domain theories have an impact beyond the computer application, and they can be shared and reused across different domain. Upper-level ontologies which embody domain theories especially have the shareable and reusability characteristics. Since upper-level ontologies are developed to represent generic concepts, they can easily be adopted across different domains. Ontologies can be used to express middle-range theories which [3] describe as theories that have a much wider applicability than the situations, contexts, or cases from which they actually originate.

As with any scientific method, ontology development as a multidisciplinary theory formulation approach needs to be evaluated and validated. Since the approach is applicable in various disciplines, the evaluation and validation approach draws upon other established scientific research approaches. Without a doubt, ontological theory needs the computational implementation and testing, but its strength is in forms of consistency and validity that are internal to the theory that is tested [3]. Subsequently, there is need for stronger notions of validation, such as external validity [3] as such, ontology engineers can draw upon

validation of theory from other scientific fields especially from the domain(s) where the ontological theories are applicable.

The classic examples of implementation of ontology as a theory formulation tool is the Basic Formal Ontology (BFO), which we will describe in the next section.

2.3 The Basic Formal Ontology

Today, we live in an information-driven society where the availability of data and information offers excellent opportunities for both researchers and practitioners. However, the overwhelming amount of information poses significant challenges in terms of its accessibility, interoperability, and reusability. These challenges are as a result of researchers using different terminologies, formats, coding systems, and software to describe their research work and results [9]. Along with differences in the representation of scientific data and information, errors in logic exist in scientific data repositories [9]. The Basic Formal Ontology (BFO) attempts to solve these problems and provide a standard approach for scientists to represent their research data and findings. Ontologies offer scientists a suitable method of representing knowledge. However, the creators of BFO argue that using an upper-level ontology will be even more advantageous.

The BFO utilises the philosophical background and principles in realism, fallibilism, perspectivalism, and adequatism. The realist approach is that ontologies are a representation of reality and not of our mental (linguistic, conceptual, theoretical, cultural) representations [52, 10]. In the realist approach, we represent entities from reality independently of the human understanding of that reality, giving us a more global and standardised representation, which will otherwise differ as human understanding differs. The principles in fallibilism holds that the statements about reality that we obtain from scientific theories may inaccurate and thus subject to correction [9]. Perspectivalism support a belief that reality is too complex to be captured in its totality by a single sci-

entific theory; hence, there is more than one legitimate perspective on reality [9, 52]. Adequatism maintains that entities in a domain can exist in different forms and at varying levels of granularity [10]. Adequatism is the opposite of reductionism, in which philosophers popularly believe that we explain complex phenomena by reducing them to smaller and more fundamental components [9].

BFO is an upper level ontology that offers a good starting point for ontology engineers to build domain ontologies through categorisation of entities and their relationships in a domain [9]. BFO is deliberately very generic and as such cannot address the specific knowledge representation needs of a particular domain, but when several scientific domains use the BFO as an upper-level ontology, it supports interoperability of data across the different domains. In order to use the BFO for ontology development, we need to understand the concepts of universals and particulars, and the structure of the BFO continuants, BFO occurrents, and ontological relations in BFO. In this section, we will discuss these concepts, the structure of BFO, and give an example from the literature of an ontology that uses BFO.

2.3.1 The universals and particulars

We have discussed that scientific theory is concerned with representing generic terminologies rather than specifics, which is the same concept applied in BFO. In section 2.1.1., we described classes and types as the abstraction of real-world entities versus instances of the classes, which are the actual real-world entities descriptions. BFO applies these notions as universals, which refer to the generic or abstraction of real entities (classes) that scientific research studies, and particulars, which are the actual entities (types) in a scientific study. BFO and the domain ontologies that extend BFO aim to represent universals as classifications of particulars and provides a hierarchy of such classifications to support reasoning over the particulars [9]. The starting point in BFO is to capture the real nature of entities involved in scientific research. The developers of BFO classified entities into two categories, continuants and occurrents. Continuants

are entities that continue to exist independently of time while occurrents are those entities that happen and may not exist at a certain point in time [9]. To sufficiently describe and model reality, researchers consider the continuants and occurrents to be complementary and co-exist. For example, there are people (continuants) having surgeries (occurrents) performed on them by other people (continuants) [9]

2.3.2 The BFO continuants

The main characteristic of continuant entities is that they persist through time and are wholly present at each moment of their existence [42]. Examples of entities that are continuants are people, characteristics of people such as height, weight, eye colour, and a place people go to work, where they live, or school. Although a continuant may lose some part of itself, at each point in time, it exists wholly. For instance, a person may cut their hair but still exists wholly even with shorter hair. BFO models different classes of continuants, which are independent continuants, generically dependent continuants, and specifically dependent continuants. Figure 2.2 shows the structure of BFO continuant with the hierarchical arrangement of its various subtypes.

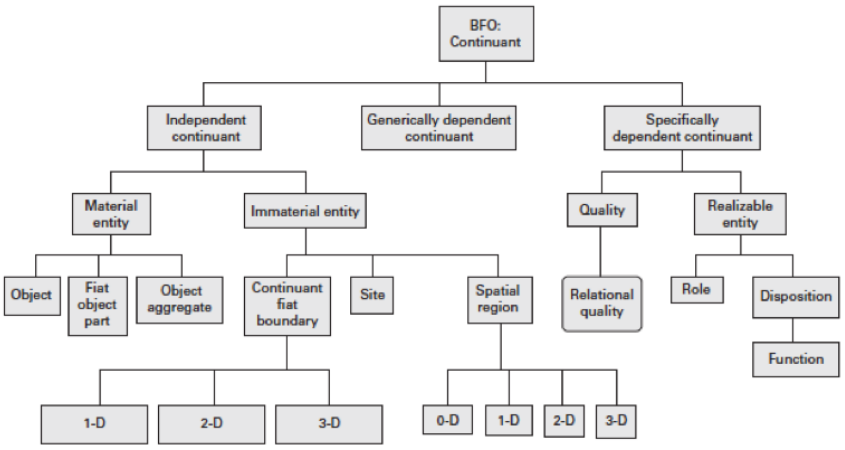

Figure 2.2: The structure of BFO continuant [9]

In BFO, the independent continuant models entities that are the bearers of

dependent continuant entities such as qualities [9]. The independent continuant is the most concrete thing that can exist, is three dimensional, and consist of material parts. There are two kinds of independent continuant entities, the material and immaterial entities. The class of material entity consists of object, parts of the object, as well as aggregate of object. Examples of members of the class of material entity can be a car (object), car wheels or engine (fiat object part), and a collection of cars in an office parking lot (object aggregate). The existence of the material entity from independent continuant cannot be based on the existence of any other entity, although it provides the basis which determines the existence of dependent continuant [9].

The immaterial entity contains the subtypes continuant fiat boundary, site, and spatial region. The continuant fiat boundary is a boundary of some material entity that exists precisely where that object meets its surroundings or the boundary of some immaterial entity such as a site. Examples of continuant fiat boundary include a surface of a chair (material entity) or the area in a laboratory where bio-hazard experiments are allowed (site). The site is an immaterial entity that exists because of some material entity and can act as a container for some other material entity. Although a site exists because of some material entity which it is defined in relation to, it does not contain this material entity as its part [9] but rather is seen as a hole or space contained within the material entity [9]. An example of a site is the empty space inside a cabinet drawer which we use to hold a stack of papers (object aggregate/material entity). This space or hole (site) exists because of the existence of the cabinet drawer (material entity), so we define it in relation the drawer, but the drawer or its components (top, bottom, sides) do not form part of this site. The spatial region is an immaterial entity subtype that is a part of space which cannot move but provides a space through with material entities move or processes occur [9]. In reference to the theory of relativity, spatial regions are defined relative to a frame of reference, for example, the frame of reference may be latitude and longitude to define a spatial region (or series of them) for a ship sailing across

the sea, or the workbench in a laboratory where an experiment occurs, or the parking lot where a car is located. For ontology engineers using the BFO, they can define their spatial region based on a frame of reference that is suitable for their ontology domain and context.

In BFO, the specifically dependent continuant models those entities whose existence are wholly dependent on the existence of independent continuant entity. Dependent continuants exhibit existential dependence in the sense that, in order for a dependent continuant to exist, some other entity in which it inheres (intuitively, an entity enjoying a larger degree of concreteness) must exist also [9]. For example, a quality like the colour of a car cannot exist without the existence of the car. Figure 2.2 shows the subtypes of the specifically dependent continuant, quality and realizable entity. The quality is the most common kind of this category of entity with examples of the class members such as colour of car, weight of a person, and temperature of water. Realizable entity represents characteristics which models role or disposition, for example, the role of a person (independent continuant) as a teacher or the disposition of a student (independent continuant) to gain knowledge.

We have established that dependent continuant requires the existence of an independent continuant to which it can inhere. To build on this explanation, specifically dependent continuant inheres to one bearer (an independent continuant) and even if the entities are similar, they cannot inhere to different bearers. For example, My suntan is specifically dependent on me. It cannot also be your suntan, however closely similar the two distinct instances of the suntan type might be. [9]

In contrast, the generically dependent continuant can migrate from one bearer to another. The generically dependent continuant is a continuant that is dependent on one or other independent continuants that can serve as its bearer. [9] An excellent example of generically dependent continuant is as the representations in the Information Artifact Ontology (IAO). The IAO provides a framework for representing information entities such as documents and dig-

ital images and their metadata [123]. An example of a generically dependent continuant will be a representation captured in a digital image of the Mona Lisa, which is dependent on a particular copy of the digital image on a hard drive (first independent continuant) and at the same time can be dependent on a different copy on a separate hard drive (second independent continuant). By having multiple digital copies of the Mona Lisa on different locations, it does not infer that there are more than one Mona Lisas and as such can have the same generically dependent continuant (the representation captured in the image).

2.3.3 The BFO occurrents

In BFO, occurrents represent those entities that are not constant in time but rather happen or evolve. The occurrent entities are never fully present at any given moment in time, but instead unfold themselves in successive phases, or temporal parts. [9] Figure 2.3 shows the structure of the BFO occurrent class with the hierarchical arrangement of its subtypes. The BFO occurrent has four subtypes, which are; process, process boundary, spatiotemporal region, and temporal region.

In simplified terms, a process is an occurrent entity which happens to some material entity or is carried out by the material entity. The existence of a process is dependent upon one or more material entity and unfolds through time [9]. For example, the process of reading a book, which requires someone or something (material entity) to do the reading (process) of a book (another material entity). A process has temporal parts, which means that the entity does not fully exist at any one point in time but instead unfolds along its temporal parts. Based on our example, the process of reading a book does not fully exist at any one point in time, but it is something that unfolds gradually for instance, over hours, days, or even weeks. This is unlike the entity book which exist entirely at any point in time because it is a continuant entity. The process of reading a book is a simplistic example, but a more complicated process will be earning a university degree. The process of earning a degree can occur over

a series of temporal parts which we can describe in different ways, for example, the different semesters, the different kinds of work expected such as projects, coursework, examination. Processes can also consist of other processes as parts, which can be proper temporal parts or temporally coextensive with the primary process and will each have their temporal parts [9]. Unlike continuant entities, if a process loses any of its part, it will no longer be the same entity.

A BFO process has process boundary, which is an occurrent entity that defines the beginning and end of the associated process. The process boundary is the instantaneous temporal boundary of a process, which is also a temporal part that itself has no temporal parts [9]. BFO does not explore the nature of process boundary and definitions of level of granularity.

The spatiotemporal region is an occurrent entity that is part of spacetime in which occurrent entities can be located [9]. Since processes occur in temporal parts, the spatiotemporal region defines the container that hold these series of temporal parts of a process. For example, the spatiotemporal region for the process of reading a book is the spacetime where the reading of the book occurs. BFO processes occur in a spacetime and are temporally extended continuum or a spacetime worm, which stretches out in and through the single unified container that is the entirety of spacetime. [9] The spatiotemporal region is three dimensional and represents a processes time, duration, beginning, and end.

The temporal region is an occurrent entity that is a part of time and serve as boundary along temporal dimensions [9]. BFO does not specify a frame of reference for any particular temporal coordinate system. Thus, ontology developers can choose their appropriate frame of reference. An example of a standard temporal coordinate system is the clock and calendar system for keeping track of time.

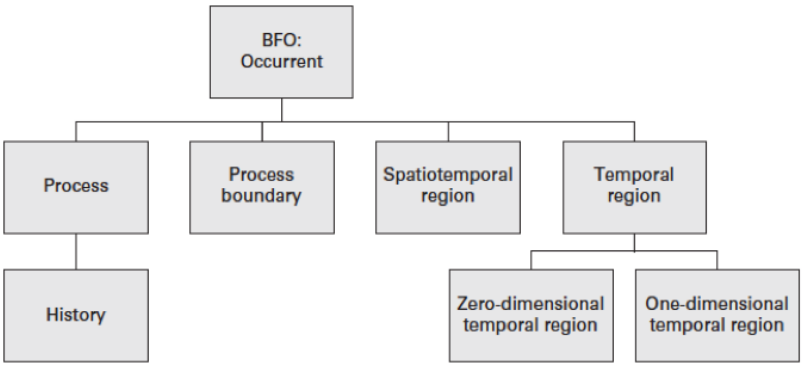

Figure 2.3: The structure of BFO occurrent [9]

2.3.4 The ontological relations in BFO

Relations in BFO allow linking of entities within the BFO and also within an ontology that extends the BFO. The relations in BFO model the basic relations that exist in reality, for instance, concepts that explain instantiation, identity and parthood [9]. There are three categories of relations in BFO, which models the relationship between two universals, a universal and a particular, and between two particulars. Some examples of relations in these categories are as follows; however, note than these examples are not exhaustive of BFO relations:

- Universal to universal: the relations in this category are present in the ontology itself [9]. There is the is_a relation, which defines an identity relationship between universals of continuants or occurrents. The is_a relation is commonly used to define an inheritance relationship between classes, and it applies to either occurrent to occurrent or continuant to continuant. For example,

  _:Lion(continuant) is_a _:Cat(continuant).

  and

  _:EatingMeat(occurrent) is_a _:CarnivorousBehaviour(occurrent).

  There is also parthood relation in this category as well as particular to particular category. In this category, the parthood relation defines the

relationship between an object and its fiat object parts in the continuant class.

- Particular to universal: the most common type of relation in this category is the instantiation relation, which describes the relationship between particulars and universals by showing that a particular entity is a member of a universal class. For example,

  _:Simba(particular) instance_of _:Lion(universal).

- Particular to particular: In this category, there is the second type of parthood relation which shows the relationship on an instance level. The parthood relation behaves differently depending on whether it is between continuants or between occurrents. Therefore BFO differentiates the parthood relation into continuant_part_of and occurrent_part_of.

Other examples of relations in the BFO include; spatial relations (located_in and adjacent_to), temporal relations (derives_from and preceded_by), and participant relations (has_participant) [9]. The participant relation shows that a continuant entity participates in the action or process, which an occurrent entity defines. For example,

_:Reading(occurrent) has_participant _:Student(continuant).

The efforts in BFO for defining relations will help in standardising how ontology developers define the relationship between the entities in their various domain ontologies, which is a challenging and confusing task. The most common relation in the literature is the subsumption relation, which consist of the identity relation (defined by is_a in BFO). This subsumption relation includes relations that allow class hierarchy definition and is often misused [55]. To address the need for a disciplined way of using subsumption, [55] proposed a formal ontology of properties. Combining the recommendations in [55] and the BFO relations will improve the proper use of relations within domain ontologies.

2.3.5 Using BFO in domain ontologies

Often, ontology developers find it challenging to understand how to begin organising and structuring terminologies during the development of domain ontologies. BFO provides a good starting point for organising terminologies, and since the BFO upper classes and relations are based on the theory of existence of entities in reality, it will be compatible with most domains. There are several examples of domain ontologies which extend the BFO; these include the Ontology for General Medical Science (OGMS) [115], Alzheimer Disease Ontology [81], Cell Ontology [14], the Foundational Model of Anatomy [111], and Ontology for Biomedical Investigations (OBI) [13].

We have previously discussed the capability of Protégé, a popular ontology development environment, to support reusing ontologies by importing existing definitions. Ontology developers who wish to use BFO as upper-level ontology can import the latest OWL version into the Protégé tool and then define their domain-specific terminologies as subclasses of the BFO classes.

2.4 Semantic technologies for the biomedical domain

Researchers are making efforts towards innovative technologies to support biomedical processes. Although we do not claim to exhaustively discuss the semantic technologies targeted at the biomedical domain, in this section, we wish to highlight a few related works. The relevant research contributions we will discuss include ontologies for the biomedical domain, examples of semantic applications for processing medical text, and some state-of-the-art technologies in the clinical laboratory.

2.4.1 Repositories of biomedical vocabularies

In recent years, the use of biomedical technologies for research and healthcare provision is becoming increasingly common. Standardised systems such as the Unified Medical Language System (UMLS) [23] and the BioPortal [94] provide valuable resources in terms of tools for processing biomedical text.

The UMLS was developed by US National Library of Medicine (NLM) as a repository for biomedical vocabularies from various subdomains, ranging from clinical terminologies to models of organisms, and biomedical literature [23]. In an attempt to promote interoperability of biomedical and health information systems, the UMLS integrates and distributes over 2 million terminologies from more than 60 families of biomedical vocabularies [23]. Some examples of vocabularies in the UMLS include NCBI taxonomy, Gene Ontology, MeSH, and SNOMED-CT [23]. The UMLS consists of a set of files and software which provides features to provide access to linked biomedical vocabularies and support the development of biomedical information systems. As part of the UMLS tools, there exists the Metathesaurus, the Semantic Network, the SPECIALIST Lexicon and Lexical tools, and the MetaMap. We will focus our discussion on the Metathesaurus and the MetaMap (see section 2.4.3).

The Metathesaurus is the main component of the UMLS and is a biomedical thesaurus that organises and links concepts from different vocabularies. The Metathesaurus is not a standardised vocabulary but rather a tool for maximising the usefulness of standardised vocabularies [116]. One of the functions of the Metathesaurus is to group terms from different vocabularies with the same meaning into concepts. The UMLS organises its knowledge base through the Metathesaurus concepts [23]. Concepts are linked to each other through relationships; these relationships either exist already in the source vocabularies or the Metathesaurus editors define them [23]. In the Methathesaurus, concepts are categorised based on 135 existing high-level Semantic Types to distinguish between different possible meanings [23, 116]. Some examples of the Semantic Types include Disease or Syndrome and Pharmacological Substance [116]. A

concept can hold more than one meaning, thereby making it belong to more than one Semantic Type. The main goal of the UMLS is to support retrieval and integration of information from biomedical sources [23]; however, there exist challenges such as ambiguity in the meaning of concepts (where concepts have more than one meaning). The UMLS provides interactive tools that allow users to find the meaning of the desired ambiguous name [116]. [110] also reported on approaches to resolving ambiguity while mapping free text to the Metathesaurus, which may be as a result of synonyms, abbreviations, or ambiguity in the meaning of the Metathesaurus concepts.

There are several examples of projects that have processed free-text to identify Metathesaurus concepts. For example, [127] processed abstracts from MEDLINE to find phrases and match them to Metathesaurus concepts across a broad spectrum of Semantic Types. UMLS provides a tool, the MetaMap which maps biomedical text to Metathesaurus concepts or discover concepts in biomedical texts (see section 2.4.3). In section 2.1.2, we briefly mentioned the BioPortal as an ontology search tool. The BioPortal is an open repository that allows users to browse, search, and visualise biomedical ontologies through Web services and Web browsers [94]. The BioPortal provides features that enable community participation by allowing users to add mappings between terms and provide comments and reviews on ontologies [138]. Like the Metathesaurus, the Bioportal is also dedicated to biomedical vocabularies and provides mappings (links) between the terms, but it does not group similar terms from across ontologies into concepts. The BioPortal contains vocabularies that also exists in the UMLS Metathesaurus [112] such as LOINC and SNOMED CT. The biomedical vocabularies and ontologies contained in the UMLS and BioPortal repositories provide some fundamental resources for natural language processing of biomedical text and development of intelligent medical applications. In the next section, we explore some of these ontologies and assess the extent to which they capture clinical laboratory procedures.

2.4.2 Ontologies for the biomedical domain

The Open Biomedical Ontologies (OBO) exists as an umbrella body for ontology developers in the domain of life sciences [121]. The OBO aims to improve interoperability of ontologies to enable better integration of data in the life sciences domain. To achieve this, the OBO provides a set of guiding principles which specify that ontologies in the OBO must be; made open and available for use without any restrictions or licence, receptive to modification, orthogonal, syntactically correct, and must use a universal system of identifiers [121]. The OBO Foundry is an initiative within the OBO which prescribe an additional set of principles (available from http://www.obofoundry.org/) to support the preceding principles and strengthen interoperability of OBO ontologies.

As part of this research work, we developed an ontology for clinical laboratory procedures. Therefore, we identified similar ontologies from the OBO that focus on biomedical procedures. These ontologies are the BioAssay Ontology (BAO) [2], the Ontology for Biomedical Investigations (OBI) [13], and the ontology for Experimental Actions (EXACT) [125]. BAO describes information about drug discovery and chemical probe screening assays and their results to categorise assays [2]. BAO aims to provide a common reference standard to support integration, aggregation, retrieval, and analysis of drug discovery data [2]. To ensure compatibility with other relevant biomedical ontologies and, BAO adopts several approaches, such as using the BFO as an upper-level ontology and reusing external ontologies like PATO, IAO, and ChEBI [2]. Several collaborative projects have successfully applied BAO, for example, in the BioAssay Research Database (BARD) [64] and for annotation of biological assays.

OBI represents different phases and activities involved in biomedical investigations and addresses the need for a cross-discipline ontology by describing terms that apply to both biomedical and technological domains. OBI provides a standardised terminology for describing experiments which facilitates comparison, reproduction and analysis and also support data exchange and information retrieval [28]. Like the BAO, OBI also makes use of BFO as the

upper-level ontology and imports parts of external ontologies. There are real-world applications of the OBI where the ontology is used to model experimental processes entities and their relations and also for annotation [28]. Instances of OBI application includes the neuroscience experiment [75, 28], vaccine protection investigation [28], and an automated functional genomics investigation [71].

The ontology EXACT provides a generic semantic representation of experimental protocols to ensure their reproducibility by humans and machines [125]. EXACT models experimental actions from biomedical protocols and all the information necessary to carry out the actions. EXACT also uses upper classes from the BFO and references several existing ontologies that have already defined the components of experimental protocols such as OBI. The framework presented in [125] can serve as a reference model for translating biomedical protocols, which exist in natural language, into machine-readable format.

The research we present in this thesis used the reference model prescribed in [125] and extended it to develop the SmartSOP framework for processing natural language clinical laboratory SOPs into machine-readable format. The OBO ontologies we describe in this section deal with biomedical experiments and investigations, but they do not describe the protocols that are specific to clinical laboratories. Therefore, there is a need for an ontological representation that fully captures the clinical laboratory procedures and the information available in laboratory protocols. This research work aims to address this need, and since the ontology EXACT is the closest to capturing the knowledge in the clinical laboratory, we extended it to develop OCL-SOP (see chapter 3).

2.4.3 Natural language processing tools for biomedical text

Semantic technologies, which consist of ontologies and the general infrastructure to generate smart and connected data, are increasingly becoming significant assets. One approach to achieving smart data is by annotating free-text documents with vocabularies in ontologies, which allows the machines to efficiently process

and use information from the documents in a multitude of ways. The free-text documents are created in natural language for consumption and use by humans. There are various kinds of free-text documents in the biomedical domain, for example, clinical notes written by doctors in a hospital during consultations with patients and experimental protocols which provides information to laboratory scientists on how to carry out experiments. In this section, we discuss several applications that are specifically designed to process free-text biomedical documents by annotating with standardised vocabularies, thus increasing the interoperability of health and biomedical information systems. We examine such applications as they are related to the SmartSOP framework we present in this thesis in terms of their basic functionality of creating machine-readable versions of natural language documents. We learnt lessons from the experiences of researchers during the development of these applications. The applications we present here all have a similar architecture to the SmartSOP framework, where an ontology (or a network of ontologies) provides a data model for the annotation of the free-text document

The first application we considered is the MetaMap, which was developed at the NLM to map medical text to the UMLS Metathesaurus or identify Metathesaurus concepts from medical text [7]. MetaMap uses a knowledge intensive approach based on symbolic, natural language processing (NLP) and computational linguistic techniques [7]. The MetaMap is a highly configurable program that allows users to set options on the output and the behaviour of the program such as how to handle word variants, common words, and word order [7]. Over the years, the functionalities of the MetaMap have become richer with the addition of features like detection of author-defined acronyms/abbreviations, browsing the Metathesaurus for concepts, detection of negation, word sense disambiguation, and chemical name recognition [8]. The pipeline components of MetaMap starts with the lexical/syntactical analysis process, which involves: (1) tokenisation and sentence boundary, acronyms/abbreviations identification, (2) part-of-speech tagging, (3) lexical lookup, and (4) syntactical analysis. The

next phases after the lexical/syntactical analysis are (5) variant generation, (6) candidate identification, (7) mapping to the UMLS, (8) word-sense disambiguation, and (9) output generation in various formats [8]. There are various ways of accessing and using the MetaMap with the most convenient being through the Java Web API. MetaMap Lite is the implementation of the basic MetaMap functions in Java language, which provides a lightweight version of the application [36].

An example of a project that has used MetaMap to extract information from medical text is [31], who extracted clinical conditions that are relevant for the diagnosis of lower respiratory infections from emergency department reports. [31] compared the performance of MetaMap with that of a physician who manually annotated the clinical reports. Several other studies have evaluated the performance of MetaMap, for example [107]also compared the performance of MetaMap with that of people. [107] found that MetaMap could identify most concepts which are in the UMLS and even identified some concepts that people did not. There are some evaluations of MetaMap that consisted of comparisons to other biomedical text NLP tools such as the clinical Text Analysis and Knowledge Extraction System (cTAKES) [109, 36] and Yale cTAKES Extensions (YTEX) [101].

Mayo Clinic developed cTAKES as an open-source NLP tool for extracting information from free text in electronic medical records [114]. cTAKES is built upon the Unstructured Information Management Architecture framework and OpenNLP toolkit [114]. cTAKES accepts either plain text or XML (compliant with clinical document architecture) as input into its pipeline and processes it through the following components: (1) sentence boundary detector, (2) tokeniser, (3) normaliser, (4) part-of-speech (POS) tagger, (5) shallow parser, and (6) named entity recognition annotator, which includes status and negation annotators [114]. cTAKES maps the named entities to concepts found in SNOMED-CT and RxNORM [114]. One technical challenge with cTAKES which was reported by [101] is that sometimes the dictionary lookup annota-

tor (named entity recognition annotator) was unable to distinguish different concepts sharing the same lexical tokens. Yale solved this issue through the development of an extension for cTAKES, the YTEX, by adding a sense disambiguation component [101]. YTEX simplifies feature extraction, experimentation with various feature representations, and the development of both rule and machine-learning based document classifiers [49]. The use of cTAKES is versatile, and an example is a project that customised it to extract UMLS medical concepts from medical reports in German language [19].

Another application for processing medical text is the Medical Text Extraction, Reasoning and Mapping System (MTERMS), which utilises ontologies to annotate natural language biomedical documents [145]. MTERMS encodes medical text using a variety of standard terminologies such as RxNorm and SNOMED and generates a machine-readable output in XML format [145]. In addition to encoding medical text, MTERMS also establishes a dynamic mapping between the different terminologies it uses. Since clinical information is often encoded using different terminologies, the dynamic mapping in MTERMS is useful for improving interoperability and integration of data from different systems [145]. The developers demonstrated how MTERMS successfully processes medication information from outpatient clinical notes in an ambulatory electronic health records system. In [144], the authors demonstrate how MTERMS can be used for mappings between different terminologies to allow interoperability. They used MTERMS to create and maintain a mapping between RxNorm and Partners Master Drug Dictionary (MDD) at both term and concept levels [144]. MDD is a local medication terminology from Partners health care system in Boston, Massachusetts.

[126] presents KneeTex, an ontology-driven NLP system that is used to extract information from MRI reports of the knee. The MRI reports consists of a narrative report in natural language, which describes the findings from the MRI scans of the knee. KneeTex adopted and expanded the Taxonomy for RehAbilitation of Knee conditions (TRAK) [30] as its knowledge base for the information

extraction. KneeTex carries out the information extraction task to identify two main kinds of entities, the finding (clinical manifestation like disease or injury) and the anatomy (the part of the human anatomy affected by the finding), as well as their qualifiers which provide more information about the entities [126]. KneeTex takes free text MRI reports as input, analyse it, and produce a machine-readable output as JavaScript Object Notation objects, which are then mapped onto the TRAK ontology to create structured and coded information [126, 30]. The structured information from KneeTex allows machines to carry out sophisticated analysis such as complex search and analysis.

We have previously mentioned the BAO as an ontology that describes terminologies for drug discovery and chemical probe screening assays. In order to allow scientist to use the BAO to create new assays effectively, [33] propose a framework to create bioassay templates, which will make content generation easier. The bioassay framework consists of a bioassay template data model, a software tool that allows experts to create and modify templates, and a standard assay template which uses the terminologies from the BAO [33]. The developers aim to make the project a community effort. Therefore they made all the resources for the bioassay template available as open-source on GitHub (see http://github.com/cdd/bioassay-template).

Literature is abundant on different approaches for biomedical text natural language processing. In this section, we explored several examples, and for this research, we considered most of these approaches and adopted suitable ones for the development of the translation engine component of the SmartSOP framework (see chapter 4).

2.4.4 Clinical laboratory technologies

Clinical laboratory technologies are increasingly becoming popular for supporting various laboratory functions, from simple procedures such as test ordering to complex procedures like the mass spectrometry. In this section, we will discuss some examples of laboratory technologies targeted at the clinical labs.

[15] discuss the adoption of Computerised Provider Order Entry (CPOE) systems in the clinical laboratory and mention some of their limitations. CPOE electronic order systems automates the laboratory test request process, which speeds up the turnaround time and integrates the process with the hospital-wide electronic records system [15]. Some of the limitations of the CPOE systems are that they do not automate the actual testing procedure or record information about the procedure and implementation of the systems have significantly high costs associated.

The clinical laboratory also adopts mobile applications such as the mobile-based e-learning application for pathology students [45], a microscopy application which uses the built-in camera in a mobile phone [27], and reference applications for laboratory results [136, 61]. Mobile applications are cost-effective, which makes them particularly useful for clinical laboratories in developing nations where obtaining funds for laboratory consumables and learning materials is difficult. However, one limitation of the mobile applications is that the information presented about the laboratory tests, such as the reference values, are not based on a standardised terminology. This makes it challenging to adopt the applications across laboratories because different laboratories typically adopt different terminologies. Another limitation of the mobile applications is that the laboratory technicians cannot use the existing applications to record test results. The laboratory scientists can use the mobile application to support some parts of the testing, such as checking reference values, but then they will have to record the results either manually or on a separate laboratory records system.

In the clinical laboratory, there are more recent attempts at technologies that encode laboratory observation information using standardised terminology such as the Logical Observation Identifiers Names and Codes (LOINC) [46]. However, there is still room for improvement in order to address the gaps in the laboratory technologies which we have identified as inadequacies of the existing technologies for supporting the testing procedure, lack of standardisation of the information

provided in the applications, and unavailability of test results recording feature integrated into the mobile applications. We also explored biomedical text NLP tools such as MetaMap and cTAKES and considered the possibility of using those to process clinical laboratory procedures free-text documents. However, the UMLS, which provides the knowledge source for both tools, do not contain some of the key concepts relevant to clinical laboratory procedures, precisely the experimental actions. These NLP tools are also not designed to identify if crucial components that are necessary for carrying out the experimental actions are present or not. The functionalities of MetaMap and cTAKES has evolved a great deal over several years, and the tools handle common NLP challenges such as dealing with negations and handing ambiguity effectively. We can learn a great deal from the implementation of both systems in designing the natural language processing tool, which will be more suited for identifying clinical laboratory procedure concepts. In this thesis, we present the SmartSOP framework, which addresses these gaps we have identified from the literature.

2.5 Summary

In this chapter, we discussed the theoretical background of this thesis and presented relevant works from the literature on which we based the proposed framework in this research. For the theoretical background, we discussed ontologies and how they can be used for both theory formation and knowledge representation. We also described the ontology development methodologies, languages, tools. Furthermore, we discussed the issue of knowledge sharing through ontology reuse and explained the BFO as an upper-level ontology that can facilitate ontology interoperability. Lastly, we mentioned some examples of ontologies from the biomedical domain, which is the domain of interest in this thesis, some examples of applications for processing biomedical text, as well as clinical laboratory technologies.

Chapter 3

Ontology Development

In this chapter, we present the development of a formal model, the ontology for clinical laboratory SOP (OCL-SOP). We demonstrate that OCL-SOP effectively addresses our first research question, which is *how can we formally represent the knowledge within clinical laboratory SOPs to allow for a standardised representation?* OCL-SOP allows us to standardise the representation of knowledge contained within SOPs for clinical laboratory procedures. This formal model is the outcome of research objective 2 and one of the main contributions of this research. OCL-SOP is one component of the proposed framework for providing IT support to clinical laboratory procedures, which we will discuss in chapter 4. The ontology provides a good foundation for the development of intelligent systems that can support procedures in the clinical laboratories. For example, robots for automation of laboratory procedures can use the knowledge about procedures that exists in OCL-SOP. In chapter 5, we demonstrate how a proposed robotics system for malaria test in the clinical laboratory can use the knowledge in OCL-SOP.

For the development of OCL-SOP, we followed recommendations from the Open Biological and Biomedical Ontology (OBO) Foundry [121] and reused representations from existing ontologies. We reused the ontology EXACT [125] in its entirety, making it the foundation for the development of OCL-SOP. The

author of this thesis was not part of the research team that developed the ontology EXACT. However, we obtained permission and support from the team to reuse the ontology.

We presented parts of the work in this chapter at the Joint Ontology Workshops in Italy in 2017 as a paper, "An ontology for clinical laboratory standard operating procedures." [79]

The rest of this chapter is structured as follows: in section 3.1, we describe NeOn methodology as the development approach for OCL-SOP. In section 3.2, we explain the lifecycle we adopted for development of OCL-SOP. In section 3.3, we describe the initiation phase and the requirements specification activity we carried out in this phase. In section 3.4, we explain the reuse phase and describe the ontology EXACT along with clinical laboratory SOP documents, which are our primary sources for the knowledge acquisition activity. In section 3.5, we explain how we re-engineered the ontology EXACT and describe the resulting structure of OCL-SOP. Finally, in section 3.6, we describe the verification activity we carried out to ensure that OCL-SOP has satisfied all the requirements we specified at the beginning of its development.

3.1 OCL-SOP development approach

We adopted the Network Ontology (NeOn) methodology for the development of OCL-SOP. The NeOn methodology framework aims to address the need for an ontology development methodology that meets the requirements of ontology engineers who are interested in reusing existing ontologies [129]. In recent years, reusing ontologies is becoming increasingly popular as a result of the availability of a large number of ontologies. There are various ways in which ontologies can be reused, from reusing selected terms to reusing entire ontologies. Also, ontology engineers may wish to re-engineer the ontologies to fit their specific application needs and purpose. NeOn methodology provides a flexible workflow that allows developers to follow different pathways for developing ontologies.

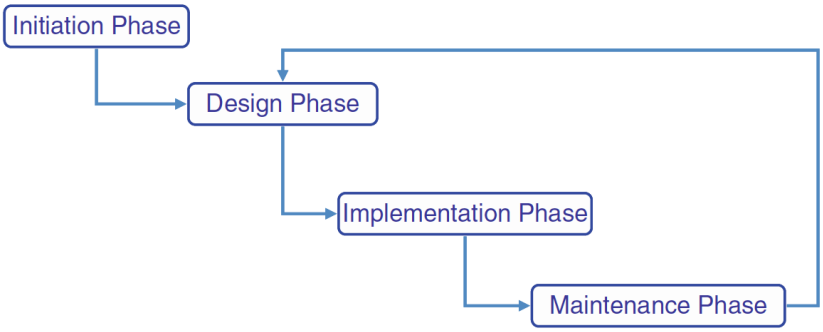

Figure 3.1: NeOn four-phase waterfall model

The NeOn methodology framework proposes a set of nine scenarios which "cover commonly occurring situations, for example, when available ontologies need to be re-engineered, aligned, modularised, localised to support different languages and cultures, and integrated with ontology design patterns and non-ontological resources, such as folksonomies or thesauri" [129]

Also, NeOn methodology provides a glossary of activities and processes which need to be carried out during the different phases of the ontology development. NeOn methodology proposes two life cycle models, the waterfall model and the iterative-incremental model. In the waterfall model, the phases of the ontology development are sequential, where one phase ends before the next one begins with no backtracking between the phases [129]. There are five versions of the waterfall model, four-phase waterfall model, five-phase waterfall model, five-phase waterfall model + merging, six-phase waterfall model and six-phase waterfall + merging model. The stages of the four-phase waterfall model start from the initiation phase to the design phase, implementation phase and maintenance phase, with the addition of reuse phase in five-phase model and reuse and re-engineering phase in the five-phase model. Figure 3.1 shows the four-phase waterfall model. The iterative-incremental model allows ontology engineers to organise ontology network development projects as a series iterations with each iteration having a set of phases which follow any of the waterfall model configurations [129].

We have extensively discussed alternative ontology development methodologies such as METHONTOLOGY [41], Cyc methodology [76], and Uschold and King's methodology [135] in chapter 2. Each of these ontology methodologies have their strengths and weakness, but most importantly, the nature of the ontology development project informs the right choice of methodology. METHONTOLOGY supports the development of ontologies from scratch or by reusing (wholly or partially) existing ontologies [41]. Cyc methodology supports automated knowledge acquisition and is suitable for extracting knowledge from large corpus of knowledge which may be difficult to encode without the support of automated tools [76]. Although METHONTOLOGY also supports ontology reuse, the entire framework of NeOn is based on reusing existing knowledge sources, both ontological and non-ontological, and gives options for different pathways of development.

Our initial investigations show that there exists the ontology EXACT, which partially fulfils the requirements of our ontology. Therefore we decided to reuse the ontology EXACT along with other ontological and non-ontological resources in the development of OCL-SOP. This decision is in line with the recommendations from the Linked Data initiative to reuse as much as possible available knowledge sources that model the knowledge needed [22]. In this regard, we adopted the NeOn framework for the development of OCL-SOP as it strongly supports reuse of existing knowledge source and provides precise guidelines for creating vocabularies.

3.2 OCL-SOP development lifecycle

For the development of OCL-SOP, we adopted the six-phase waterfall model, which consists of the initiations, reuse, re-engineering, design, implementation and maintenance phases. The six-phase waterfall model is suitable for a situation which fits NeOn frameworks scenarios 3 and 4, which deals with reusing and re-engineering ontological and non-ontological resources. Since we have al-

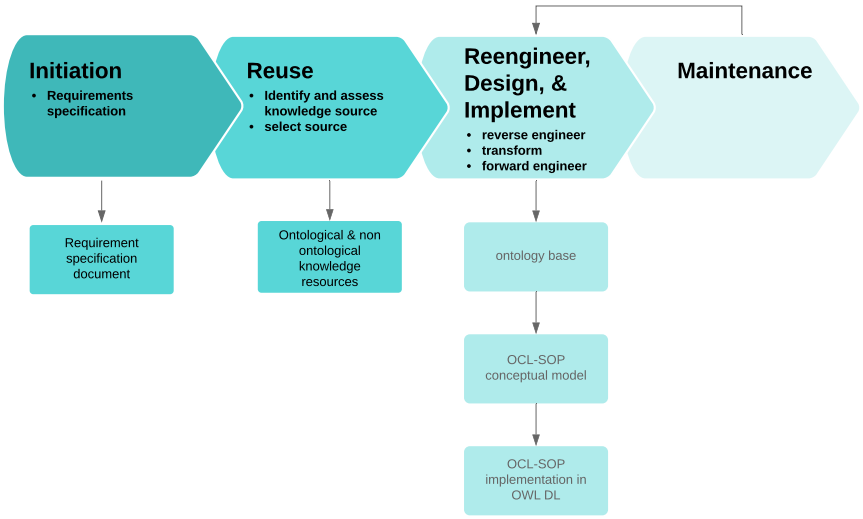

Figure 3.2: OCL-SOP Lifecycle Model

ready identified the existence of at least one ontological and one non-ontological resources from our initial investigations, we conclude that the six-phase waterfall model is the best fit for our scenario. Figure 3.2 shows an outline of the ontology development lifecycle model for OCL-SOP with the different activities, processes, and outputs involved in each phase.

In sections 3.3 to 3.5, we explain how we accomplish each of the activities from the initiation to the implementation phases. In this thesis, we will not report on the maintenance phase, but the activities in the phase will continue to happen even after the completion of this research work. The maintenance phase requires revising and updating the ontology if errors occur during its use, leading to the creation of new versions [129].

3.3 Initiation phase

The core activity in the initiation phase is the requirement specification which consists of steps taken to determine the requirements the ontology is expected to fulfil. The requirement specification activity is critical as it allows identi-

fication of the kind of knowledge that the ontology needs to model, allows a focused search of the knowledge sources, and is useful for verification of the ontology at the end of the development [131]. The set of tasks we carried out during the requirements specification activity are to identify the purpose, scope, implementation language, intended users, intended use and functional and non-functional requirements. We then created a list of competency questions from the functional requirements. Competency questions are defined in natural language which an ontology should be able to answers. Gruninger & Fox stated regarding competency questions that "ontology must contain a necessary and sufficient set of axioms to represent and solve these questions" [54]. These competency questions are helpful in the verification of ontology. We carried out the requirement specification tasks through a series of interviews and focus group discussions with domain experts from the clinical laboratory. We also carried out observations at two clinical laboratories to understand how procedures are carried out and the differences that can occur across laboratories. The output of the requirements specification activity is the ontology requirements specification document shown in figure 3.3.

The ontology requirements specification document facilitated the reuse phase, where we searched for existing ontological and non-ontological knowledge sources for reuse and the verification of OCL-SOP at the last phase.

3.4 Reuse phase

We have established the need to reuse both ontological and non-ontological knowledge sources for developing OCL-SOP in line with the principles of Linked Data initiative. In the NeOn methodology framework, the reuse process consists of a set of similar activities for both ontological and non-ontological knowledge sources. These activities are: searching for knowledge sources, assessing the set knowledge sources, comparing ontologies (only for ontological knowledge sources) and selecting the most appropriate knowledge source. We used the

OCL-SOP Ontology Requirements Specification Document	
1	Purpose
The main aim of developing OCL-SOP is to provide a knowledge model for clinical laboratory standard operating procedures.	
2	Scope
The ontology focuses on describing the knowledge for procedural actions carried out in the laboratory.	
3	Implementation Language
The language of implementation is OWL-DL	
4	Intended Users
User 1: Clinical laboratory scientists carrying out actions in the laboratory	
User 2: Clinical laboratory domain experts developing and revising SOPs	
User 3: Other medical personnel receiving test results from laboratory	
User 4: Automation engineers desiging automated systems for the clinical laboratory procedures	
5	Intended Uses
Use 1: Looking up information regarding laboratory procedures in the clinical laboratory	
Use 2: Publishing new SOPs	
Use 3: Looking up test results from the clinical laboratory	
Use 4: Building machines that can read and process clinical laboratory SOPs	
6	Ontology Requirement
Non-Functional Requirements	
NFR1: Representing standardised terminology for clinical laboratory procedures	
NFR2: Reusing existing knowledge sources	
NFR3: Writing and publishing the ontology following the linked data principles and best practices	
Competency Questions from Functional Requirements	
CQ1: Which experimental actions are involved in a particular clinical laboratory experimental procedure?	
CQ2: Which data actions are involved in a particular clinical laboratory experimental procedure?	
CQ3: What descriptors are mandatory for an action?	
CQ4: What descriptors are optional for an action?	
CQ5: What methods are available for an action?	
CQ6: Which actions have methods?	
CQ7: What are the synonyms of an action?	
CQ8: What is the temperature range for an experimental action?	
CQ9: What is the period range for an experimental action?	
CQ10: What are the conditions for an action?	
CQ11: Which materials are required for an action?	
CQ12: What is the speed for an experimental action?	
CQ13: What is the volume for a material?	
CQ14: What is the concentration value for a biochemical entity?	
CQ15: What laboratory finding is the output of a data action?	

Figure 3.3: A fragment of the requirements specification document

information in the ontology requirements specification document as a guide to search for the most appropriate knowledge sources.

We already had access to the ontology EXACT which partially fulfils the requirement of our ontology. However, we still searched for online sources to find similar ontologies. Based on the scope of the ontology to focus on the representation of clinical laboratory procedure actions, we determined that the National Center for Biomedical Ontology (NCBO) [90] is a good starting point to search for relevant ontology. We searched through the NCBO Bioportal but could not find any ontology that sufficiently represents clinical laboratory procedures. Therefore we decide to reuse and re-engineer the ontology EXACT. Section 3.4.1 provides a description and structure of the ontology EXACT.

We also searched for non-ontological sources and found a collection of SOPs from the department of Public Health England (PHE) [108] that represents clinical laboratory procedures. We discussed the non-ontological knowledge source with domain expert and established a consensus that the PHE SOPs along with hospital-specific lab SOPs forms an adequate knowledge base for OCL-SOP. We identified 47 PHE SOPs and 30 laboratory-specific SOPs for use as the ontology knowledge base. In section 3.1.3.2, we describe the SOPs we chose to extract terminological knowledge for clinical laboratory procedures.

3.4.1 Structure of the ontology EXACT

The ontology EXACT represents "the full semantics of biomedical protocols required for their reproducibility" [125]. The motivation for the development of the ontology EXACT arise from the "need for the better representation of biomedical protocols to enable other agents (human or machine) to better reproduce results" [125]. Several previous versions of EXACT exists. For example, there is EXACT/EXPO, which is suitable for automated laboratories and EXACT/OBI, which is more suitable for use within OBO communities [124]. However, we chose the latest version of the ontology EXACT [125] for reuse in the development of OCL-SOP.

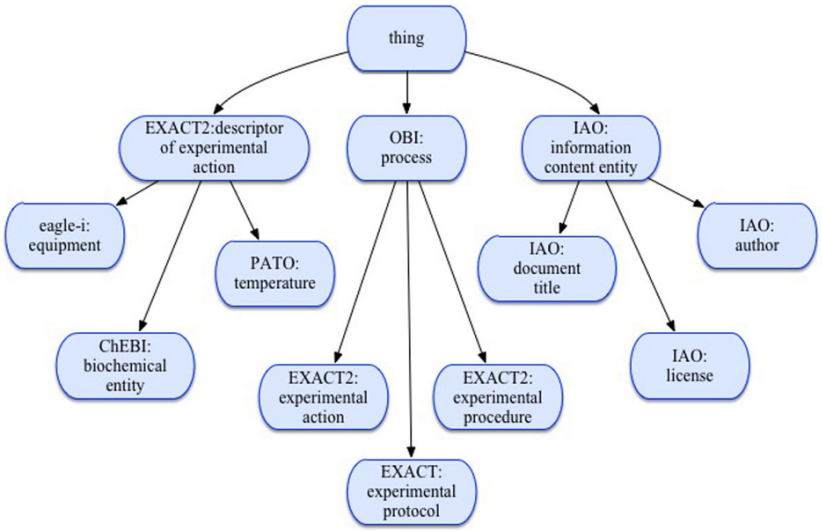

Figure 3.4: Structure of the ontology EXACT [125]

The structure of the ontology EXACT is simple, and it reuses classes from the upper-level ontologies BFO, IAO (the Information Artifact Ontology), PATO (Phenotype And Trait Ontology) and OBI [124]. Figure 3.4 shows an overview of the upper classes of EXACT. EXACT uses the IAO class 'information content entity' to model information about SOPs as textual content. The information in the 'information content entity' are relevant but are not actual actions or descriptors of actions, for example, notes and authors. EXACT reuses the OBI class process, which models processual entities and consists of the EXACT classes' experimental action', 'experimental protocol' and 'experimental procedure'. The 'experimental action' class models actions carried out during experiments to achieve some goal. EXACT also models all the information required to carry out the experimental actions in the 'descriptor of experimental action'. Certain experimental actions require specific information and "failure to record such essential information may result in the failure to correctly follow biomedical procedures, and produce erroneous results" [124]. The descriptor class has as subclasses some reused terms, for example, equipment from eagle-I and temperature from PATO. The descriptor class represents all the properties of an action to capture essential information. For example, "an action 'incu-

bate' requires the description of a 'period' of incubation and what' biochemical entity' will be incubated" [79]. EXACT also models some information that may be useful but are not mandatory to complete a process as '(optional) descriptor of experimental action'.

In EXACT, a set of relations link the processual entities to their descriptors and the information content entity. Some of the relations in EXACT are from the OBO Relations Ontology (RO), which link real-world physical entities that BFO models [124]. The ontology EXACT imports several relations from OBI, for example, 'has quality' and 'has participants'. EXACT defines new relations such as 'has proposition' to link physical entities to information entities [124]. The inverses of all the relations exist in EXACT, which are useful for defining relations between entities in reverse.

3.4.2 Clinical laboratory SOPs

As previously mentioned, we identified as our knowledge base, 47 PHE SOPs and 30 laboratory-specific SOPs. PHE commissions the development of several standard documents among which are the UK Standards for Microbiology Investigations (UK SMIs). The UK SMIs are "are a comprehensive referenced collection of recommended algorithms and procedures for clinical microbiology" [108], which are accessible through a public repository on the UK government website. At the time of this research, there were 47 available SMIs which contained procedure steps, therefore we downloaded all of them. The SMIs consist of these categories: bacteriology, virology, identification, test procedures and quality-related guidance [108]. We selected the SMIs from all the available categories. These SMIs are developed in natural language by domain experts, who follow a rigorous process to ensure quality and accuracy in information. Importantly, the SMIs represent knowledge that is agreed upon by experts in the field, which makes it a suitable knowledge source our ontology. The National Health Service (NHS) in the UK recommends that clinical laboratories adopt the prescribed UK SMIs to enable standardisation and sharing of best prac-

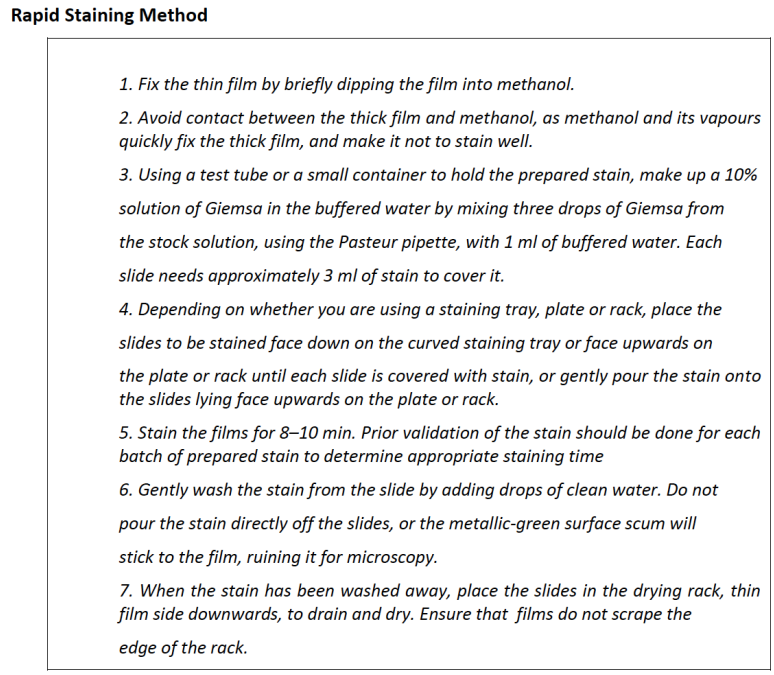

Figure 3.5: SOP for rapid staining method for malaria microscopy test [96]

tices. Laboratories also develop their own SMIs, which take into consideration the type of equipment available and local practices. However, getting access to the lap specific SMIs is not easy. We contacted 5 laboratories in the UK and 10 in Nigeria to request access to these SMIs but only obtained 30. During the evaluation, we obtained 10 additional SMIs which we used to measure the efficiency of the SOP translator (see section 4.3.4). Figure 3.5 is an example of protocol document showing the rapid staining method for malaria microscopy test which we obtained from the National Guideline for Diagnosis and Treatment of Malaria in Nigeria. We also provide an example of the complete UK SMI in Appendix A.

Discussions with our collaborators in the clinical laboratories reveals that individual laboratories can either develop their own SOPs or adopt those provided by standards organisations such as the UK SMIs. Our collaborators provided 30 laboratory-specific SOPs for our analysis and knowledge acquisition activity. Like the UK SMIs, these SOPs also exist in natural language

3.4.3 Knowledge acquisition activity

We carried out the knowledge acquisition activity throughout all the phases of the ontology development, but we completed the majority of this activity in the reuse phase. We started by analysing and manually extracting clinical laboratory procedure terms from the PHE and hospital-specific SOPs. We sorted the terms into categories of either experimental and data actions or descriptors of those actions. We then compared the experimental and data actions to the experimental actions in EXACT. We realised that although EXACT contains some of the actions, there are several that were missing. Figure 3.6 shows an example of this analysis, where we compared all the actions from the Catalase Test Procedure in the PHE SOP to EXACT to identify actions that exists in EXACT and those that are missing. In this example, the actions Place and Pick were present in EXACT but Rub, Cap, Tilt and Observe are missing. We also found some of the EXACT descriptors of experimental actions insufficient for representing the descriptions of the clinical laboratory SOPs. The outcome of this activity was a list of actions we needed to include in the new ontology and structural changes to EXACT that are necessary to adequately represent the clinical laboratory SOPs. During the rest of the OCL-SOP development phases, we continuously searched through the Bioportal to find ontologies that already define the actions and descriptors that are missing in EXACT. We then reused those terms and their definitions. We only defined new actions in OCL-SOP if we could not find a suitable representation in another ontology.

3.5 Reengineer, design and implementation phase

We carried out the reengineer, design and implementation phases together as some of the activities in these phases are overlapping. We used the ontological and non-ontological resources from the reuse phase and reengineered them to create the OCL-SOP. The goal of reengineering non-ontological resource is to convert that resource into an ontology through a series of activities. The ac-

Experimental/Data Action	Sentence	In EXACT?	Synonym	Descriptors
Place	Place 4 to 5 drops of hydrogen peroxide solution in a test tube or bijoux bottle.	TRUE	add	Biochemical entity: hydrogen peroxide solution volume: 4 to 5 drops equipment: test tube or bijoux bottle
Pick	Carefully pick a colony to be tested with a wire/loop or disposable alternative.	TRUE	pick	biochemical entity: colony equipment: wire/loop or disposable alternative
Rub	Rub the colony on the inside wall of the bottle just above the surface of the hydrogen peroxide solution.	FALSE		biochemical entity: colony equipment: bottle biochemical entity: hydrogen peroxide solution
Cap	Cap the tube or bottle.	FALSE		equipment: tube or bottle
Tilt	Tilt it to allow the hydrogen peroxide solution to cover the colony.	FALSE		Biochemical entity: hydrogen peroxide solution biochemical entity: colony goal: allow hydrogen peroxide solution to cover the colony
Observe	Observe for immediate bubble formation.	FALSE		goal: for bubble formation

Figure 3.6: Finding the Catalase Test Procedure actions and descriptors in EXACT

tivities are: reverse-engineering the resource to create a representation at an abstract level, transforming the resource into a conceptual model, and forward engineering to create an implementation of the ontology in a formal language [129]. If reengineering an ontological resources is required, it means that the ontology is not useful in its current form for a particular use case.

The reengineering of ontological resources consists of similar activities with non-ontological resource activities, which are reverse engineering, restructuring, and forward engineering. The reengineering of an ontology can be carried out at any level of abstraction of either specification, conceptualisation, formalisation, or implementation [129]. For the design phase, the output will be a conceptual model of the proposed ontology [129], this coincides with the output of transforming non-ontological resources and restructuring of ontological resources. The conceptual model is the basis of creating a formal representation in an ontology in the implementation phase.

3.5.1 Activities

In the reengineer, design, and implement phases of OCL-SOP development life-cycle, we carried out these activities:

- Reverse engineer SOPs for clinical laboratory procedures: we carried out the reverse engineering of this non-ontological knowledge source. For the SOPs, we processed them with the guidance of domain expert and identified procedures and actions from the procedure steps. We also identified descriptors of the actions from the document. We represented the outcome of processing the SOPs in a table to form our ontology base. Figure 3.7 shows a fragment of the ontology base table which we obtained from processing the SOP for 'catalase test' procedure

- Reverse engineer the ontology EXACT: We analysed the ontology EXACT and determined that the current structure cannot adequately represent content of the ontology base or fulfil some functional requirements. The ontology EXACT fails to answer the competency questions CQ1, CQ5 CQ9, and CQ15. To address the need for changes in the structure of the ontology EXACT, we decided to reengineer at the conceptualisation level and as a result reverse engineered to this level.

- Transform and restructure: we carried out the transformation of the ontology base into a conceptual model by restructuring the ontology EXACT to create the conceptual model for OCL-SOP. We added new classes, modified some definitions, and changed some of the class hierarchies to fulfil the functional requirements of OCL-SOP. In section 3.5.2, we describe the structure of OCL-SOP, highlighting the changes we carried out.

- Forward engineer: we carried out this activity by implementing the conceptual model using OWL DL. We used Protégé as the ontology development environment to design and implement OCL-SOP. Protégé provides tools that allowed us to carry out the conceptualise and implement activ-

Sentence	Action	Descriptors
Place 4 to 5 drops of hydrogen peroxide solution in a test tube or bijoux bottle.	Place	biochemical entity, volume, equipment
Carefully pick a colony to be tested with a wire/loop or disposable alternative.	Pick	biochemical entity, equipment
Rub the colony on the inside wall of the bottle just above the surface of the hydrogen peroxide solution.	Rub	biochemical entity, equipment
Cap the tube or bottle.	Cap	equipment
Tilt it to allow the hydrogen peroxide solution to cover the colony.	Tilt	biochemical entity, goal
Observe for immediate bubble formation.	Observe	goal

Figure 3.7: A fragment of the OCL-SOP base table

ities together.

3.5.2 Structure of OCL-SOP

OCL-SOP inherits its structure from EXACT, and all the upper classes remain unchanged. Figure 3.8 shows the upper level of the OCL-SOP. In this section, we focused on describing the parts of OCL-SOP that are new additions or EXACT terminologies that we have reengineered. We structured the section based on the upper classes in the ontology. We previously mentioned that EXACT follows the principles of defining entities from the BFO, therefore in OCL-SOP, we maintained the same principles. We represent the entities that are occurrents as processes, which includes the data actions and experimental actions. We then described the continuant entities as the descriptors of experimental actions. Furthermore, we reused several of the BFO relations such as has participant.

Information content entity

The 'information content entity' remains unchanged in OCL-SOP. We maintained the subclasses that allow us to model information on textual entities such as 'alert messages', 'note' and 'author identification' and other informa-

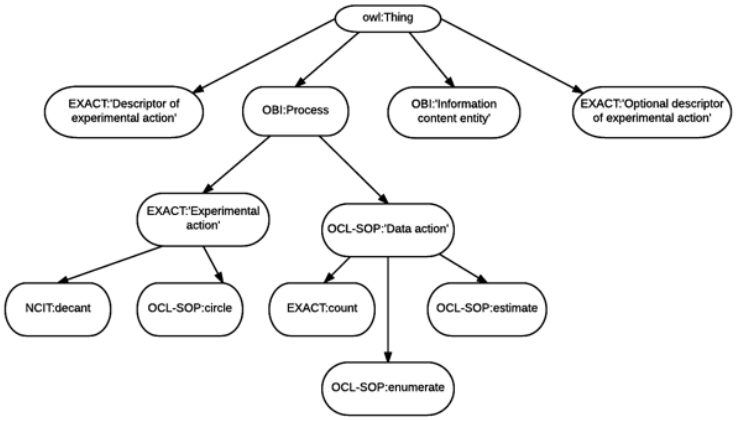

Figure 3.8: Upper classes of OCL-SOP

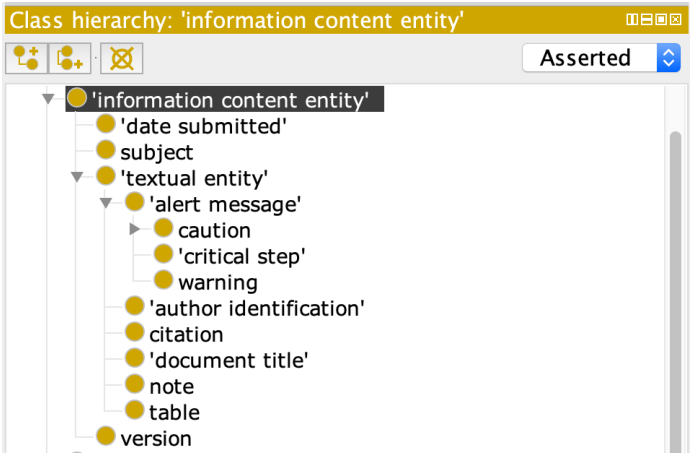

Figure 3.9: Information content entity branch in OCL-SOP

tion such as 'date submitted' and 'version'. Figure 3.9 shows a hierarchy of the 'information content entity' branch.

Process

In the OBI process class, we maintained the 'experimental action' class and added more actions as subclasses. However, the most significant change in the process class from EXACT is the addition of the 'data action' classes. Figure 3.8 shows the process class with the 'experimental action' and 'data action' branches.

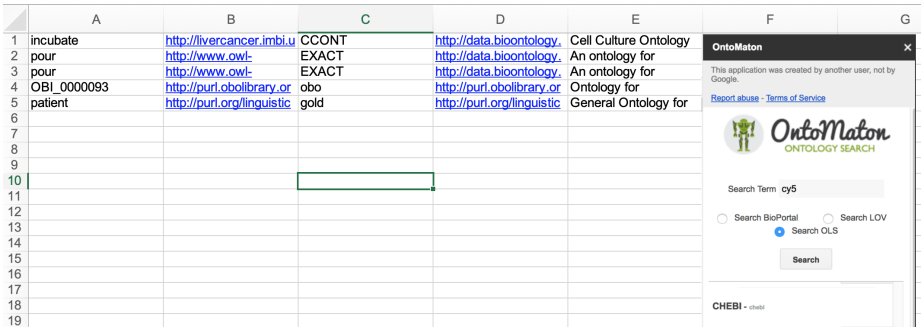

Figure 3.10: New OCL-SOP terms identified with OntoMaton

Experimental actions

The output of the knowledge acquisition activity revealed 68 new experimental actions which did not exist in any other knowledge representation definition. We added these classes to OCL-SOP as new classes. We also identified some terms that are currently in other knowledge models. For example, we found that 'decant' is defined in the National Cancer Institute Thesaurus (NCIT) [89]. To determine which terms already exists, we created a list of all the new terms identified during the knowledge acquisition activity, i.e. terms that are not present in EXACT. We then used OntoMaton [78], an ontology search tool, to find ontologies in the BioPortal that have defined the terms on the list. Following the best practices in ontology development and the recommendations by OBO Foundry [121], we reused existing definitions and imported the relevant classes to OCL-SOP. Figure 3.10 shows some examples of new terms and the output of the OntoMaton search tool. We observed that some of the terminology in SOPs differs from the terminology used the ontology EXACT. A different (synonymous) term may refer to the same experimental action. For example, 'agitate' is a term used in SOPs and is a synonym for the term 'shake' that already exist in EXACT. We added the synonyms as annotations to the relevant classes in OCL-SOP. Figure 3.11 shows an example, 'move' has several synonym annotations in OCL-SOP.

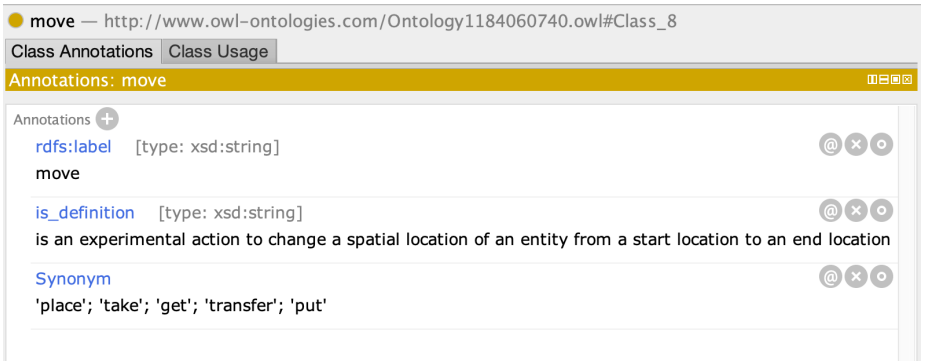

Figure 3.11: Action with several synonyms in annotation

Data actions

During the analysis of the SOPs, we identified several data-specific actions. We extended OCL-SOP by adding the 'data action' sub-branch to the 'process' branch to model data-specific actions. Figure 3.13 shows a class hierarchy of the 'data action'. In EXACT, the actions 'record', 'measure', 'calculate', and 'count' were defined as 'experimental actions'. We re-modelled these actions as data-specific actions. We also added new 'data actions', e.g. 'convert' and 'estimate'. We identified that 'data actions' behaved slightly differently from 'experimental actions'. Data actions models those actions that manipulate some data entity. Therefore, we added the 'data entity' to 'descriptors of experimental actions'. We will discuss the 'data entity' in the next section. To provide a context on how 'data actions' are used, we present some examples in figure 3.12 showing data specific actions as they are presented in the natural language SOPs.

Descriptors of experimental action

The 'descriptors of experimental action' models the properties of processes, both 'experimental actions' and 'data actions', for example, 'biochemical entity' and conditions. The processes are related to their descriptors through the 'has relation' property and its sub-properties. For example, figure 3.14 show the class description of the 'experimental action' 'add'. We made significant changes to

Data Action	Description	Sentences Showing Usage in NICE SMIs
report	same as state or record. however, context may differ if it is used as a noun rather than a verb. for example “send report to doctor’s office”, report here is not an action but an entity	e.g 1: Report susceptibilities as clinically indicated. e.g 2: Report on the actual numbers, or range of WBCs and RBCs per litre or per mL according to local protocol.
calculate	to determine a value from some formula which can be used to represent a data entity	e.g: Calculate value of one eyepiece division as follows: 10 eyepiece divisions = 0.20 1 eyepiece division = 0.20/10 = 0.020mm = 20µm
Measure	to determine a number or some other measurement of an entity	

Figure 3.12: Some data actions found in SOPs

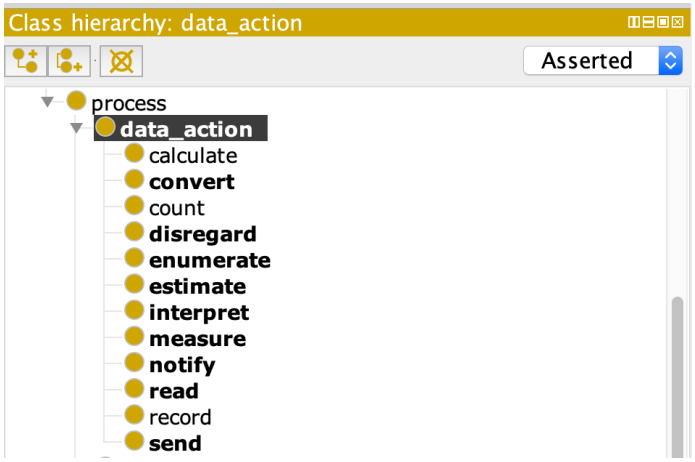

Figure 3.13: Hierarchy of the data action branch

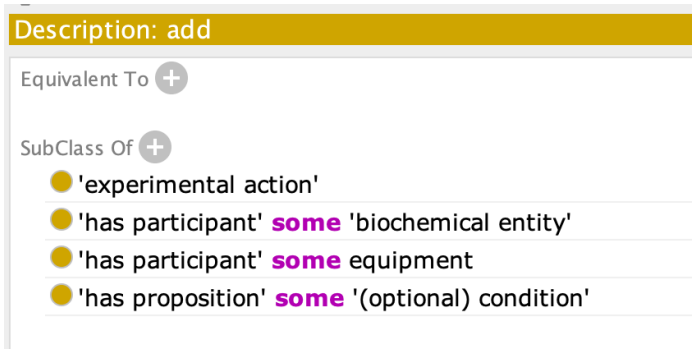

Figure 3.14: Class description of ’add’

this branch in OCL-SOP, as highlighted below:

Representing temperature and period as ranges We identified that temperature and period are sometimes presented as ranges in SOPs. For example, store at a temperature between 4-6°C. To properly model such ranges, we changed the temperature and period classes to 'min temperature', 'max temperature', 'min period', and 'max period'. In some instances, there are exact values rather than ranges for the temperature or period, so the same values for the min and max should be used. For example, in the following sentence, there is a range for period and exact value for temperature.

Incubate at 37°C for 4-5hr.

In this case, the 'min temperature' has a value of 37°C and the 'max temperature' also has a value of 37°C. While the 'min period' has a value of 4 hours and the 'max period' has a value of 5 hours.

Data item In OCL-SOP, we introduced the 'data action' branch, which models actions found within the clinical laboratory protocols that deal with or result in the generation of data items. We then created 'data item' in the 'descriptors of experimental action'. The 'data item' class exists in the Neurodegenerative Disease Data Ontology (NDDO) along with subclasses 'clinical finding' and 'laboratory finding', therefore we imported these classes into OCL-SOP. Figure 3.15 shows the class definition of data item. We related the 'data action' class to the 'laboratory finding' through the 'has specified output' relations as such

'data action' 'has specified output' 'laboratory finding'.

Protocol method During our analysis of SOPs, we found out that some processes have methods, and it is important to represent such methods as part of OCL-SOP. EXACT already has the 'protocol method' class in 'descriptors of experimental actions' so we redesigned it and added as subclasses the methods

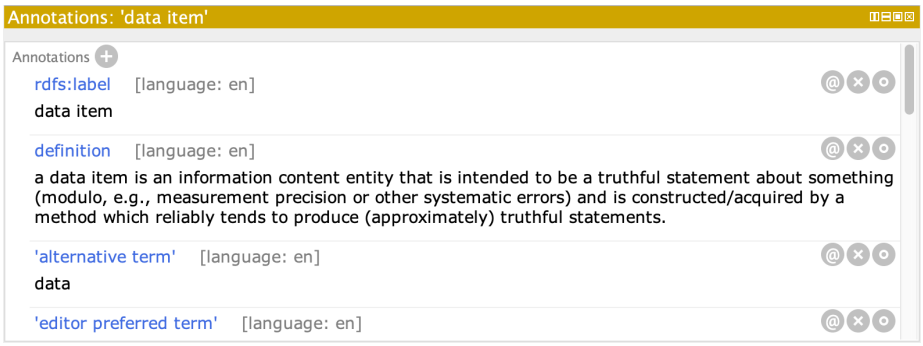

Figure 3.15: Class definition for 'data item'

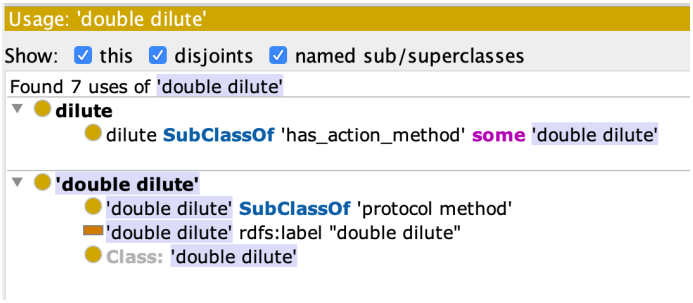

Figure 3.16: 'double dilute' method for 'dilute' action

we identified from the SOPs. For example, the action 'dry' has several methods such as 'air dry' and 'blot dry'. Some of these methods already exist but as 'experimental actions' thus we moved such methods to the 'protocol method' class. We then created the 'has action method' relation and used it to relate actions to their corresponding methods. Figure 3.16 shows an example of the 'experimental action' 'dilute' which has a method 'double dilute'.

Relations for defining descriptors of experimental actions OCL-SOP inherits all the relations from EXACT, however, none were suitable for describing the link between an action and its method. We included the relation 'has action method' to link 'process' entities to the methods of carrying out actions.

Optional Descriptors of experimental action

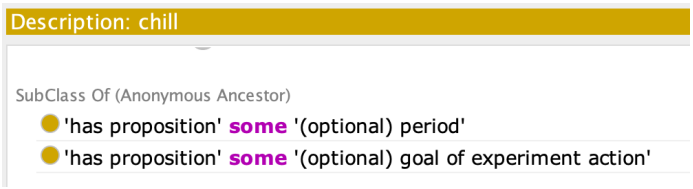

Figure 3.17: Optional descriptors for 'chill'

In EXACT, '(Optional) descriptors of experimental actions' are used to model characteristics that are not essential for an experimental action without complicating the representations in the ontology [125]. The optional descriptors are related to the process using the 'has relation' property and its sub-properties. An example of non-essential properties is 'optional period' for the experimental action 'chill' shown in figure 3.17. This shows that although it will be nice to have the value for period, the 'chill' experimental action can be successfully carried out without that information. Since the 'optional descriptor of experimental action' branch contains the same terminologies as the 'descriptors of experiment action branch', we mirrored the same changes we implemented in the descriptors here.

3.5.3 Publishing of OCL-SOP

The NFR3 we defined in section 3.3 identifies the need to publish the ontology following the linked data principles and best practices. We used GitHub to publish and for versioning of OCL-SOP. The latest version of OCL-SOP and all import files are available on GitHub, which is accessible through the ontology URI http://www.w3id.org/OCL-SOP in OWL/XML format.

We also produced a documentation for OCL-SOP in HTML using the Wizard for Documenting Ontologies (WIDOCO) [48]. WIDOCO automatically creates and publishes a rich documentation based on an ontology in OWL/RDFS. WIDOCO has features that allows the users to customise the documentation through a graphical user interface. The documentation consists of properties such as the ontology name and URI, versions of the ontology, and details of the

ontology axioms. Figure 3.18 shows a screenshot of the OCL-SOP documentation from WIDOCO. This documentation is also available on GitHub through the ontology URI.

3.6 Evaluation of OCL-SOP

The NeOn approach defines ontology evaluation as an activity, which aims to determine the quality of an ontology by comparing it against a frame of reference [129]. The frame of reference could be the set of functional requirements in the form of competency questions or real-world scenarios. Ontology evaluation consists of activities for verification and validation. Ontology verification checks the correctness of the ontology implementation in a formal language during development, and validation assesses whether the ontology accurately models the real world in the domain of interest [41]. There are various tools and techniques available for ontology evaluation activities. Most of the ontology development methodologies prescribe an approach for evaluation, for example [129] and [41]. Other approaches are: a framework for evaluation of knowledge sharing technologies, which includes ontologies [51], guidelines on how to check for inconsistencies, incompleteness, and redundancies [50], and OntoClean [56]. Several of the proposed techniques for ontology evaluation involve checking the ontology against a set of competency questions. There are also tools such as built-in Protégé reasoners [88] and the OntOlogy Pitfall Scanner! (OOPS!) [106], which automate the verification of ontologies.

For the evaluation of OCL-SOP, we carried out both verification and validation activities. We carried out the validation during the user-centred evaluation of the SmartSOP framework, which we will describe in chapter 6. We assessed the suitability of OCL-SOP for supporting the core functionalities of the SmartSOP framework by measuring how well those functionalities performed. The SmartSOP framework, which we will describe in chapter 4 relies heavily on the knowledge in OCL-SOP to translate free text SOP into machine-readable for-

mat. If the knowledge represented in OCL-SOP is incomplete or inaccurate, the quality of the output of the proposed framework will be negatively affected. We used the validation activity to check that OCL-SOP sufficiently models real SOPs in clinical laboratories.

We carried out the verification activity iteratively to identify faults in the implementation of OCL-SOP in OWL-DL and correct any issues we found. We used two approaches for the verification activity and carried out a total of three iterations. In the first approach, we used the FACT++ reasoner in Protégé to check OCL-SOP for inconsistencies in the definitions. We fixed any problems the reasoner identified and reran the reasoner until there were no problems after the third iteration. Figure 3.19 shows an example of the output of the reasoner check from Protégé. In the first iteration, the reasoner identified a total of 12 inconsistencies. Some of the inconsistencies are as a result of wrong domain or range declaration:

> for example,
>
> 'has participant' has a domain of 'experimental action' or 'data action' and a range of 'material entity'

but a declaration in OCL-SOP shows 'blot dry', which is not an experimental action or data action, using the relation. Another inconsistency that appeared is the wrongful assertion of descriptors as equivalent of process: for example OCL-SOP shows,

> 'cap' as an equivalent of 'equipment':
>
> instead, it should be
>
> 'cap' 'has participant' 'equipment'.

For the second verification activity, we used a formative evaluation approach to iteratively check if OCL-SOP meets all the requirements we set out during the requirements specification activity. Whenever a requirement failed, we made the necessary changes to the ontology and checked again in another iteration. In

the requirement specification document, we identified functional (as competency questions) and non-functional requirements (see figure 3.3). To demonstrate that OCL-SOP fulfills the set of functional requirements, we checked that the ontology can answer the competency question. We used the competency questions to check that OCL-SOP offers accurate and complete information about clinical laboratory procedures. We used the DL-Query feature in Protégé to create and run queries that we derived from the competency questions on OCL-SOP. During the first iteration, the OCL-SOP answered 8 out of 15 questions correctly. We corrected our representation in OCL-SOP and ran the queries for two more iterations until we obtained correct answers for all the questions. In figure 3.20 we show an example of how we tested the competency questions and the obtained from DL-Query. We chose the competency question "CQ6. Which actions have methods?".

language en

Ontology for Clinical Laboratory Standard Operating Procedure (OCL-SOP)

Release June 2019

This version:
http://www.w3id.org/OCL-SOP

Latest version:
http://www.w3id.org/OCL-SOP

Previous version:
https://github.com/fatibaba/EXACT-med/blob/master/EXACT-med.owl

Revision:
EXACT-Med (Experimental actions (medical)) ontology has been renamed to OCL-SOP (Ontology for Clinical Laboratory Standard Operating Procedures). The latest version can be found at http://www.w3id.org/OCL-SOP

Authors:
Fatima S Maikore

Contributors:
Dr Larisa Soldatova

Extended Ontologies:
EXACT

Cite as:
Maikore, Fatima S., Gantigmaa Selenge, Adebola Olayinka, Pamela Abbott, and Larisa N. Soldatova. "An Ontology for Clinical Laboratory Standard Operating Procedures." In JOWO. 2017.

1. Abstract

This ontology aims to provide a model for the representation of protocols in clinical laboratories. Such an explicit computational model is important for ensuring reproducibility of experimental procedures, facilitating experimental data exchange, and supporting automation. OCL-SOP extends EXperimental ACTions (EXACT) ontology by reusing all its terms and adding new ones pertinent to clinical laboratory standar operating procedures

Figure 3.18: OCL-SOP Documentation.

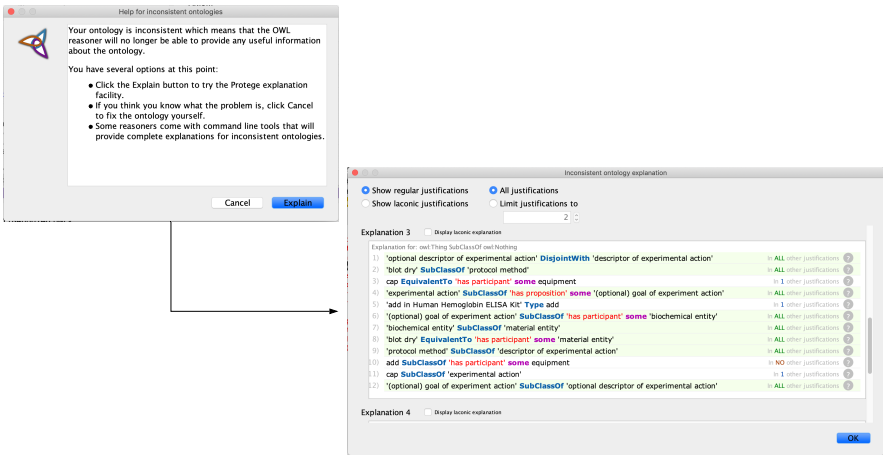

Figure 3.19: Output of Fact++ reasoner verification on OCL-SOP

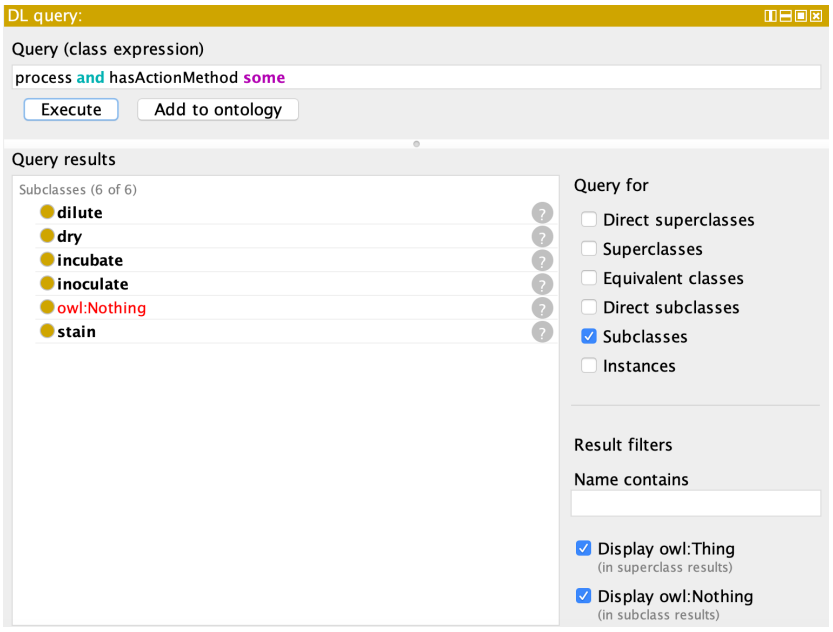

Figure 3.20: Sample result of verification with competency question

Chapter 4

The SmartSOP Framework

In the previous chapter, we presented and discussed the development of OCL-SOP as an ontological model for the representation of clinical laboratory SOPs. Ontological models can be used to support the development of intelligent systems through standardisation of knowledge sources and providing such knowledge models in a machine-readable and understandable format. We demonstrated how OCL-SOP addresses our first research question, which is:

> 1. *How can we formally represent the knowledge within clinical laboratory SOPs to allow for a standardised representation?*

In this chapter, we present our novel contribution as the SmartSOP framework for providing IT support to clinical laboratory procedures. The framework addresses our second and third research questions which are:

> 2. *How to automatically convert SOPs represented in natural language to a machine-readable format while minimising loss of essential information*

and

> 3. *How can we present the clinical laboratory SOP to lab scientists in a way that makes it easy for them to access and use information while monitoring their adherence to the guidelines?*

This framework addresses the research questions by providing novel contributions as follows:

1. The **OCL-SOP** as an ontological model to provide a standardised clinical SOP representation

2. The **SOP translator** to translate free text SOPs into machine-readable format

3. The **mobile application** to provide laboratory practitioners with access to the standardised SOPs and encourage adherence to guidelines.

We have published portions of the work presented in this chapter in a peer-reviewed journal as a paper titled A framework for IT support of clinical laboratory standards [80] and also presented a paper, An ontology for clinical laboratory standard operating procedures [79] at the Joint Ontology Workshops. The SOP translator component of the framework was originally developed by some of the co-authors of the journal paper. The previous version of the SOP translator is presented in [125]. For this research, we adopted and modified the SOP translator to process clinical laboratory SOPs. In section 4.3.1, we describe the changes we made to the structure and algorithm of the SOP translator. The mobile application component of the SmartSOP framework was also developed as a collaborative effort. An undergraduate student developed the first iteration of the mobile application as a final year project, and we were part of the supervisory team. We conceived the idea for the mobile application initially to contribute to the SmartSOP framework and invited the undergraduate student to develop the first iteration. We presented the first iteration of the mobile application in [79]. We built-on the first version of the mobile application and developed the subsequent versions. In section 4.4, we describe the latest version of the mobile application as the third component of the SmartSOP framework.

The rest of this chapter is structured as follows; in section 4.1, we explain how the framework functions and provide a high-level representation of the

different components of the framework. In section 4.2, we explained how OCL-SOP works within the framework and provided an example of a free text SOP which we used to demonstrate the functionalities of the framework. In section 4.3, we describe the SOP translator and its algorithm, show the changes from the previous versions, and present demonstration of the translator on the free text SOP. Finally, in section 4.4, we describe the SmartSOP mobile application and its core functionalities.

4.1 Description of the framework

We designed and built the SmartSOP framework to support the laboratory practices by providing machine-readable and processable SOPs and allowing practitioners to record the results of procedures. We followed the design science approach in this research, and this chapter describes the output of the design and development activity, which is the proposed framework. The design and development activity of design science research has some flexibility in the use of research strategies. This activity aims to create an artefact, and so long as it works, any approach for generating the artefact is admissible [69]. For this activity, we used a rapid prototyping approach where we designed and built each component of the framework iteratively. We tested each iteration and refined the next iteration based on the results of the testing to ensure that we capture fully all the requirements we have specified for this framework. These requirements have changed and evolved, hence the need for the continuous refinement of the prototypes. The changing requirements makes the rapid prototyping approach more suitable than other approaches where requirements need to be fully defined at the beginning of the development activity for example in the case of the waterfall model [69].

In this section, we will describe the overall framework and then provide details of the individual elements of the framework and their core functionalities. The SmartSOP framework consists of three major components:

1. An Ontology for Clinical Laboratory SOP (OCL-SOP) which provides the vocabulary required by the SOP translator

2. A SOP translator which converts natural language SOP into machine-readable format based on the vocabulary in the ontology

3. A SmartSOP mobile application which demonstrates the implementation of the framework

Figure 4.1 shows a high-level representation of the proposed SmartSOP framework. The framework accepts input from an authorised person in the form of a free text SOP. The SOP translator will then process the SOP by identifying key entities and matching them to the representations found in OCL-SOP. The SOP translator will create an output file which is the SOP in a machine-readable format. The SOP translator will also identify any missing relevant term in the SOP and add a note to the output SOP. The note can later be used by the SOP designers to improve the quality of the SOP by ensuring that they capture all the required data elements. The SOP translator can also detect relevant experimental actions that are missing from the OCL-SOP and alert the user to update the ontology. This function provides an automated way of updating OCL-SOP with the most relevant clinical laboratory SOP terminology. The machine-readable SOPs will then be stored in a secured database to be accessed later by relevant software applications. To demonstrate the usefulness of the output from the SOP translator, we developed a mobile application which takes the machine-readable SOPs and displays it in a user-friendly manner to laboratory practitioners. The mobile application provides additional complementary features such as recording test results in a shared database and an easy search functionality.

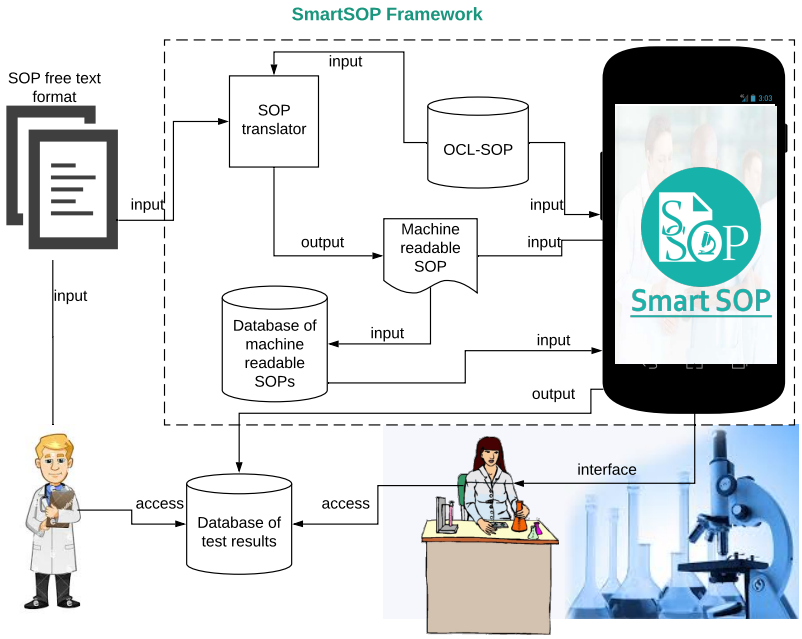

Figure 4.1: Overview of the SmartSOP framework

4.2 OCL-SOP within the SmartSOP framework

OCL-SOP serves as the knowledge model used within the framework which provides the clinical SOP terminology required by the SOP translator to process and convert free text SOPs to machine-readable format. We have described the development and structure of OCL-SOP in details in Chapter 3.

The use of ontologies as part of Natural Language Processing (NLP) tools have become increasingly popular. In recent years, researchers in the biomedical domain have adopted the approach of using NLP methods to extract and process information from clinical text. This approach offers various advantages such as improvement in the quality and amount of information available to clinicians. Previous research work has demonstrated the usefulness of ontologies for representing and retrieving semantic knowledge in a machine-readable and understandable format [70]. UMLS provides the Metathesaurus, which is a rich thesaurus of biomedical concepts obtained from ontologies such as SNOMED

CT, and serves as a knowledge source for NLP tools like MetaMap and cTAKES [116]. We used an approach similar to that of [144] where ontologies such as SNOMED and RxNorm where used to map terminologies from clinical notes. In our case, OCL-SOP provided the most convenient set of rich vocabulary for mapping terminologies in clinical laboratory SOPs. OCL-SOP was explicitly designed to model the procedures and all related information found in clinical laboratory SOPs.

In this section, we discussed the relevant parts of OCL-SOP that we used within the framework. We also describe an example of a clinical laboratory procedure and how it maps to OCL-SOP. In subsequent sections, we demonstrate how the SmartSOP framework works using the example clinical laboratory procedure described here and how the framework is used to improve the lexicon of OCL-SOP.

4.2.1 OCL-SOP Components

OCL-SOP is an ontology that models the typical actions carried out within the clinical laboratory. We described these actions in terms of how they are carried out, the conditions required, biochemical entities involved, equipment used, as well as input and output data. Figure 2 is a fragment of OCL-SOP showing the classes utilised within our proposed framework.

A typical clinical laboratory SOP contains information about the OCL-SOP classes illustrated in figure 2. The relevant classes from OCL-SOP used in the framework exist in the branches of *experimental action, data action*, and the *descriptors of experimental action*. In SOPs, there are two main kinds of actions, those contained in the *'experimental action* and those in the *data action* branches. The experimental actions are carried out in the procedure steps, which requires the laboratory technician to manipulate some physical entity such as a *biospecimen* or an *equipment*, for example, *pour* or *shake*. Some of the experimental actions are carried out under certain conditions, for example, the action *shake* may need to be carried out *vigorously*, and that becomes the

condition. Experimental actions also have certain descriptors that depicts the *quality (temperature, speed, or volume), quantity,* and *duration* of the actions and the materials (*equipment and specimen*) that are involved in the action. The *'data action'* branch consists of actions that deal with the manipulation of data element both as input and output of specific procedures. For example, a procedure may require the laboratory scientist to *record* the results or outcome of some procedures or *calculate* some of the parameters of an experimental step. By carrying out a *data action*, an output may be generated in the form of a *laboratory finding*. There are also *protocol methods* that apply to both data and experimental actions. There are slight variations in the terminologies used across different laboratories which leads to the same actions possibly having different names. We included a representation that captures common synonyms of actions in the OCL-SOP by relating the actions to a literal value through the *hasSynonym* property.

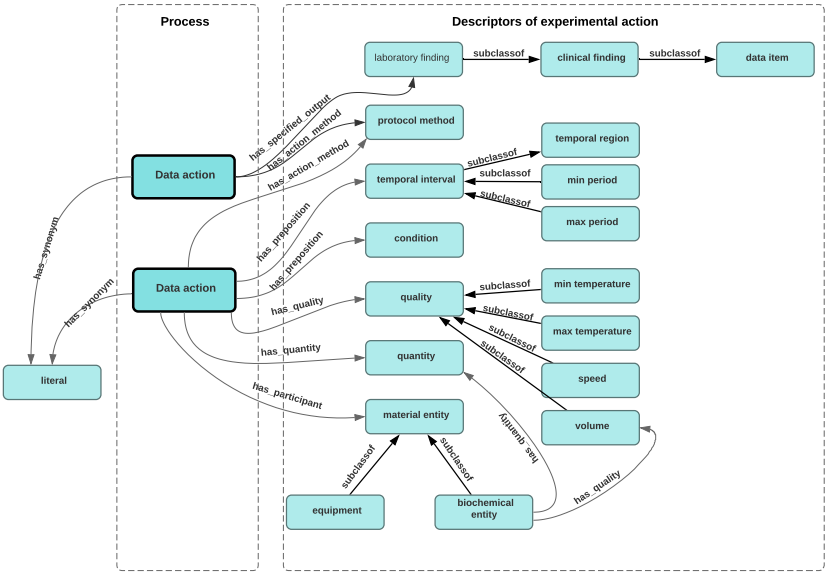

Figure 4.2: A fragment of OCL-SOP

4.2.2 Mapping the Urine Microscopy SOP to OCL-SOP

In a clinical laboratory SOP, there is a section that outlines the procedure in terms of the different actions the laboratory technician needs to carry out and describes all relevant information for the action. A typical SOP will provide various instances of the OCL-SOP classes illustrated in figure 4.2. In this section, we describe an example of a typical clinical laboratory SOP provided in natural language. We chose the Urine Microscopy SOP found in the Investigation of Urine document from the Public Health Englands collection of the UK Standards for Microbiology Investigations [108]. The procedure involves looking at urine specimen under a microscope to identify cellular components. Figure 4.3 shows a fragment of the free text urine microscopy SOP.

In order to demonstrate the compatibility of the Urine Microscopy SOP with the OCL-SOP classes, we manually mapped the descriptions in the SOP to the classes in OCL-SOP. The flowchart in figure 4.4 shows the manual mappings with all the relevant classes. For the manual mapping, we started by identifying the different actions required and arranged them in the order they are processed. The SOPs explicitly states the order of the actions. The SOP starts its description of actions from the first required action, and the subsequent actions appear in which order they are expected to be completed. We then identified all the relevant descriptors for each action and mapped them to their corresponding OCL-SOP classes. Some descriptors for an action may appear in a preceding statement rather than the statement which mentions the action. For example, in Figure 4.3, the fourth statement mentions the scan action, but one of the equipment to be used for the scanning, the inverted microscope, is mentioned in the third statement.

In the next section, we process the free text version of the urine microscopy SOP through the SOP translator and generate an output file in a machine-readable format. The SOP translator takes the OCL-SOP as one of the inputs in the algorithm for translating the free text SOPs. The experimental actions and data actions are recognised and mapped to the text from the free text SOPs

along with their descriptors.

4.4 Microscopy or Alternative Screening Methods

4.4.1 Standard

Microtitre tray with an inverted microscope

Mix the urine gently, to avoid foaming.

Using a pipette and disposable tips, dispense known volume (~60µL, see 'Note 2' below) of mixed urine to a numbered well in a flat-bottomed microtitre tray. Make sure that the specimen covers the whole bottom surface area (the use of a template will facilitate matching the specimen and well number).

Allow to settle for a minimum of 5min, but preferably 10–15min, before reading with an inverted microscope.

Scan several fields in each well to check for even distribution of cells and urine.

Count the numbers, or estimate the range, of WBCs and RBCs per representative field and convert to numbers (or range) per litre.

Enumerate and record SECs.

Figure 4.3: An example of SOP in free text, the Urine Microscopy

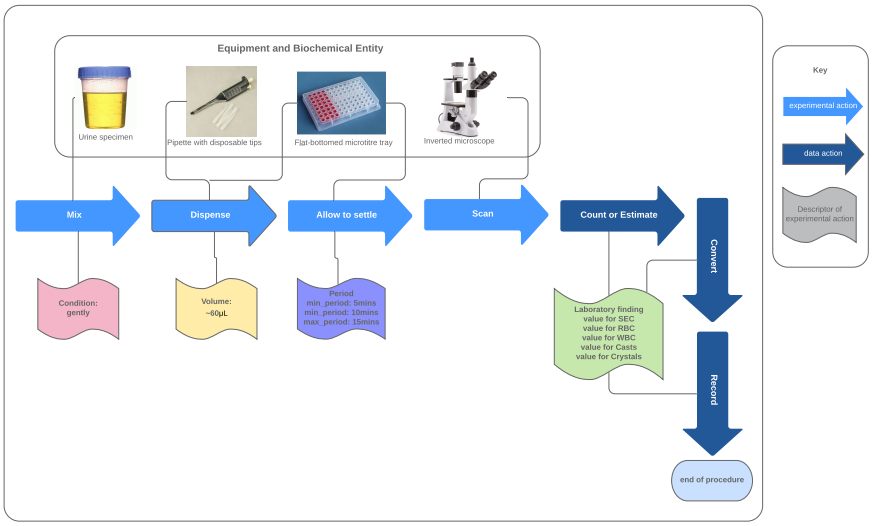

Figure 4.4: Manual mapping of Urine Microscopy SOP to OCL-SOP Classes

4.3 The SOP translator

The SOP translator is an NLP tool which processes free text SOPs and convert it into a semantically annotated machine-readable format. SOPs are typically written in natural language which makes it difficult for machines to read and understand their content [80]. This creates several problems in the clinical laboratory in terms of automation and computation of laboratory procedures and their variables as well as lack of interoperability between different laboratories.

Our proposed framework provides a SOP translator that is capable of providing the much-needed machine readable SOPs.

The SOP translator also serves as a semi-automatic tool for ontology update which is useful for improving the lexicon of the OCL-SOP. We had to analyse a large amount of protocols to identify OCL-SOP terms. This created the need to automate the process to make it easier and faster than processing each protocol manually. One problem of such an automation is that clinical laboratory SOPs exist in various formats as free text, where the terminology is often ambiguously used with different measurement units used across different laboratories [80]. We took into consideration these problems during the creation of the SOP translator.

The SOP translator uses OCL-SOP as a knowledge base to recognise the experimental actions and identify their corresponding descriptors within free text SOPs. The SOP translator recognizes specific actions that are not yet defined in OCL-SOP, which can be added manually by the user. Additionally, the missing descriptors for specific actions may be used for document verification and quality assessment of the SOPs.

In the NLP community, there are various techniques for extracting information from free text based on terminologies in domain vocabularies. Earlier works such as Barrows et al. used a methodology which combines lexical and morphologic text matching techniques and manual review by experts to extract diagnostic clinical terms from a controlled vocabulary [16]. More recent NLP works in the medical domain include Zhou et al. on the development of the Medical Text Extraction, Reasoning and Mapping System (MTERMS) for extracting and processing drug information from medical text [144, 145]. We have discussed several other examples of related works on NLP for medical text extraction in chapter 2.

We adopted and modified an existing algorithm for the SmartSOP SOP translator which uses OpenNLP tool [133] to process the free text SOP and prepare it for text extraction and mapping to the OCL-SOP terms. We will

describe the algorithm process in section 4.3.2.

4.3.1 Changes to the SOP translator

We have mentioned previously that the algorithm for the SOP translator was first developed by [125] to convert free text experimental protocols into machine readable format based on EXACT ontology. We adapted and modified the algorithm to make the SOP translator useful as the second component of the SmartSOP framework. The first major change is to allow the SOP translator to recognise terms from OCL-SOP. The first version of the SOP translator was designed to work with terms from the ontology EXACT. Since OCL-SOP differs significantly from EXACT we made other necessary changes to the SOP translator. OCL-SOP contains synonyms of experimental and data actions, therefore we added a synonym recognition module to the SOP translator. The new SOP translator can also distinguish between data action terms and experimental action terms. This is useful to allow us to easily associate the data actions with the outcome of those actions during the recording of test results in the clinical laboratory. In section 4.5, we will discuss how the mobile application records test results. In the free text SOPs, there are usually lists of materials that are required to carry out the procedures. The new SOP translator can now identify equipment and biochemical entities from the list and update the members of the material entity class of OCL-SOP. The SOPs sometimes contain more than one action in a statement, however, the previous version of the SOP translator can only detect the first action in any statement. In the new version, all the actions are successfully identified in the statements. We added a lemmatization technique to the SOP translator to allow it to detect actions from verbs that are in the continuous tense. For example, the SOP translator can detect the experimental action mix from the word mixing.

NLP is prone to a lot of ambiguity, for instance, even though the actions in OCL-SOP are action words, not all verbs translate into an action in OCL-SOP. Another example is that some words can represent both actions and noun, such

as plate representing both a noun (as equipment) and verb (as experimental action). We encountered a lot of challenges in dealing with such ambiguity and we have still not resolved some of those problems in the SOP translator.

4.3.2 The SOP translator process

In this section, we describe the SOP translator and the tasks performed by each of its parts. The SOP translator consists of the protocol tagger, a protocol parser and an output writer. Figure 4.5 shows an overview of the SOP translator. The protocol tagger accepts as an input the pre-processed SOP document and uses tools from the OpenNLP library to prepare the SOP for identification of actions and descriptors. The Apache OpenNLP Library is a tool for processing natural language processing text by performing tasks such as tokenization, sentence segmentation, part-of-speech tagging, and named entity recognition [133]. Similar tools such as Stanford CoreNLP and ANNIE from the GATE framework also carry out similar tasks and perform on similar level with the OpenNLP [105]. However, we had previous experience with OpenNLP and found the APIs easy to incorporate into our application, this informed our decision to use the tool for the SOP translator. In the protocol parser, we used an approach similar to Named Entity Recognition to identify the actions and all their descriptor values from the protocol tagger output. Finally, the output writer prepares the output file with the identified elements. The output file is a machine readable SOP document.

The protocol tagger

The SOP translator requires a set of input in order to process the SOPs. First there is the free text SOP that needs to be translated. The free text SOP is pre-processed to convert it to plain text void of any formatting styles and saved with a .txt extension.

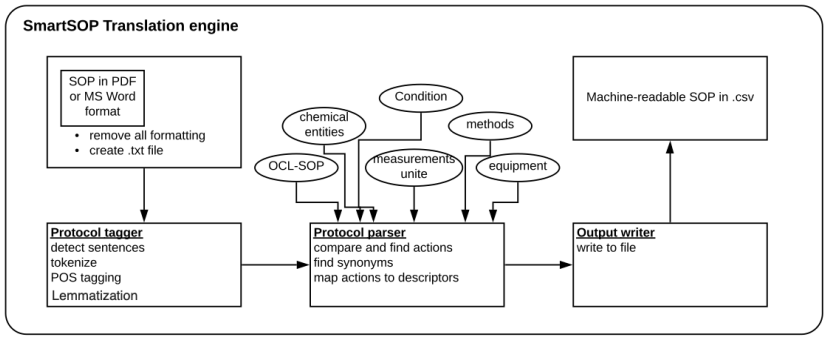

Figure 4.5: SOP translator components

Sentence detection The first step for the protocol tagger is to read the text file and identify the sentences using OpenNLP sentence detection tool. OpenNLP uses the Sentence Boundary Disambiguation (SBD) method to detect sentences by identifying the beginning and end of sentences. Usually sentences end with a period (.). However, there are cases where the period appears at places other than the end of the sentence, for example in email address or URLs (.com).

Tokenization The protocol tagger then takes each sentence and divide it into tokens. The OpenNLP tokenize tool offers three methods of breaking sentences into smaller fragments, which are: (1) the SimpleTokenizer tokenizes text using character classes, (2) the WhitespaceTokenizer divides the sentences by identifying white spaces in between the words, and (3) the TokenizerME uses Maximum Entropy to decide how to separate tokens [133]. The WhitespaceTokenizer essentially divides the sentences into words while TokenizerME can be trained to identify word phrase [133]. We chose the WhiteSpaceTokenizer because the actions and descriptors we are trying to identify most often occur as single words rather than phrases therefore there is no need to train any model.

Part-of-speech tagging The protocol tagger uses the tokens for part-of-speech (POS) tagging to tag each token with its corresponding part of speech.

OpenNLP provides a predefined model, the en-posmaxent.bin, which is trained to tag tokens with their corresponding word type such as noun singular or mass (NN), verb base form (VB), verb past tense (VBD), etc [133]. OpenNLP also allows the user to train the POS tagger model with a training material which is a set of tokenized sentences with a POS tag already attached [133]. Training the model will be useful if the tokens are a set of word phrases or special technical words but since our tokens are single words, we used the pre-trained model provided by OpenNLP.

Lemmatization In the earlier version of the SOP translator, used the tokens that are tagged as VB to map to the actions from OCL-SOP. However, this approach meant that the translator was missing some actions that are presented differently in the text due to inflection and derivation. For example, actions written as "mixing" and "separation" are missed by the translator. We addressed this problem by including lemmatization on the tokens that have already been tagged with POS. An alternative approach will be to use stemming, however, this method targets to remove the derivational affixes and leaves the word stem [1]. Whereas lemmatization will use a vocabulary and carry out a morphological analysis of the words to remove the inflectional endings and return the lemma or base word [1]. We focused on inflectional changes to words in the SOP text. We decided on lemmatization because it works well on identifying inflectional words and the resulted lemmas does not require further processing and we are able to use them in the protocol parser directly. This allowed us to capture actions that were earlier missed by the translator due to inflectional forms of the action words.

OpenNLP offers two approaches for lemmatization, statistical, which requires a lemmatizer model to be trained and a dictionary based method [133]. We adopted the dictionary based method because the words in the SOP text, which are important for identifying actions, are common words. There is an OpenNLP dictionary available for use which contains all possible combina-

tion of common words with their corresponding lemmas. The protocol parser identified the lemmas for all the POS tagged tokens from the dictionary en-lemmatizer.dict.

The protocol parser

The lexicon of the SmartSOP framework consists of a subset of terms from OCL-SOP, which provides the actions, their corresponding descriptors and synonyms for mapping with the actions found in the SOP. The protocol parser uses the technique of named entity recognition (NER) to identify the actions and descriptors from the SOP text.

OpenNLP provides several pre-trained models used to identify named entities such as person, location, time, and organisation from text [133]. We however, built our training models to capture the relevant named entity types for this application. Table 4.1 shows the named entity types for the protocol parser.

Named entity type	Description
Action	action classes from OCL-SOP with the associated descriptor classes
Equipment	classes of equipment used for procedures in lab
Chemical	chemical names and IDs
Measurement unit	units of measurements for values of entities such as volume or temperature
Methods and conditions	methods and conditions associated with actions

Table 4.1: Named entity types

The training model for the action named entity provides training data with a list of action classes from the OCL-SOP along with their associated descriptor classes. We have a separate model that provides entity type for identification of the types of equipment mentioned in the SOP text. We obtained a list

of required equipment for the different lab procedures, which are usually lab specific, and are therefore provided by the lab or found within the specific SOPs. We used the equipment list to train the equipment entity type model. Additional training models include the measurement units and IDs, procedure methods, and procedure conditions.

The translator also requires a list of chemical entities which is used to identify the chemical entities that might occur in the text. The chemical entity type model uses training data from a selection of chemical entities and the unique chemical IDs from the database of Chemical Entities of Biological Interest (ChEBI) [38] and PubChem [134]. However, chemical names can be very ambiguous with different clinical laboratories using different terminologies. To address this challenge, we are currently collecting different chemical names from laboratories and from the SOP text and using those to train the model further. However, we still have challenges in matching the right IDs to the chemical names, especially for the chemical names we obtained from the SOP text and from the labs. [139] mentioned that NLP tools do not work well on chemical nomenclature and therefore proposed a lexical and two statistical approached to improve NLP capabilities for processing chemical text. [8] attempted to solve this problem by adopting the approaches proposed by [139] to include chemical name recognition in MetaMap.

We created the algorithm to carry out the NER and the pseudo-code is shown in Algorithm 1. The steps in the algorithm are described below:

- The protocol parser reads each sentence from the SOP text and for each sentence that is eligible for processing, it compares all the lemmas from the POS tagged tokens to the NER model for actions to identify the actions in the sentence.

- Since there could be more than one action in a sentence, the protocol parser treats each matched action separately. For each action found, the parser greats a token grouping that contains the lemmas in a sentence

Algorithm 1: Algorithm for SOP translator protocol parser

```
initialization;
while there is a sentence in file do
    if this is the first sentence then
        Then set the first sentence as the title of the SOP;
    else
        if sentence starts with caution, note or warning then
            Do not process;
            Go to next sentence;
        else
            Check all lemmas from protocol tagger against the action and
             synonym type model to find entity names for the action type;
            if actions or matching synonyms are found then
                while there is an action do
                    Create a new token grouping from the position of
                     action in sentence till the end;
                    Process each token grouping separately to map to
                     corresponding descriptor;
                    if descriptor is found among tokens then
                        map biochemical entity to chemical id;
                        map measurement to measurement unit;
                        map methods and conditions;
                    else
                        Add note descriptor not found in SOP;
                    end
                end
            else
                if token has a POS tag of verb then
                    add note action not found in model, need to update
                     OCL-SOP with action;
                else
                    add note "no action found in statement";
                end
            end
        end
    end
end
```

from the position of that action to the end of the sentence. Figure 4.6 shows an example of sentence with three actions, count, estimate, and convert, and how the parser breaks down the sentence.

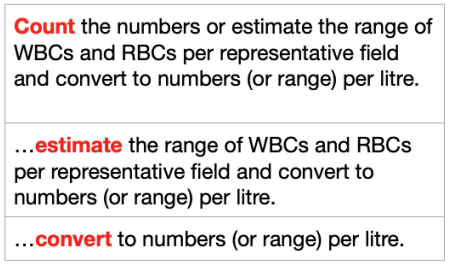

Figure 4.6: Parser breakdown of multi-action sentence.

- The parser then checks each token grouping to identify the descriptors of the action by matching to the NER models for descriptors. In most cases, the descriptors are mentioned after the action is named in the text, however there are few instances where this is not true.
- If descriptors that matched the named entity types of biochemical entity or entities with a measurement unit (volume, temperature, etc), then the parser will map the biochemical entity to the chemical id and the measurement entity to the corresponding measurement unit.
- If action entity or descriptor entity is not found, it means that either:
  - the action and/or relevant descriptor is missing from trained models and hence need to be updated. The protocol parser adds a note in the output file to indicate what term is missing and needs updating. Since the SOP translator is also being used as a semi-automatic ontology update tool, this scenario illustrates how we use it to discover new terms.
  - Or the relevant descriptor(s) is missing from the free-text SOP and the protocol parser adds a note in the output file stating that a key descriptor not available. An authorised person in the clinical

laboratory can use this information to update the SOP document and provide all essential descriptors. This allows us to improve the quality of the SOPs by ensuring all relevant information for carrying out a laboratory procedure is available.

The output writer

After processing all sentences in a given SOP, the output writer of the SOP translator generates an output file which is both machine and human readable. This output file is a comma separated value (csv) file which can easily be viewed as a table with columns showing the actions and various descriptors found in a free text SOP. Table 4.2 shows the names and descriptions of the columns in the table for the output file. Each row in the output table represents a sentence from the SOP, the identified actions and their corresponding descriptors. Empty cells in the output table means one of two things; either the action does not require that particular descriptor, or that descriptor has not been found in the text. The latter case is documented and attached to the Notes section of that particular row. Beyond that, the SOP translator also documents the cases where the extracted action is not found in OCL-SOP and thus the ontology can be updated with the new actions.

The output files are stored on a dedicated NoSQL database server and can easily be converted into other machine-readable formats such as RDF or XML. This provides flexibility on how the machine-readable SOPs can be used to support operations such as search, retrieval, comparison, sharing, and versioning. We demonstrated an example of this application with the SmartSOP mobile application which reads and processes the output file from the SOP translator.

Named entity	Description
Statement	The sentence for processing
Action	The action identified in a statement
min Temperature	the lower limit temperature

Named entity	Description
max Temperature	the upper limit temperature
Volume	a volume with its unit of measurement
Concentration	a concentration value with its unit of measurement
Equipment	the unique identifier for the equipment
min Period	the lower limit period
max Period	the upper limit period
Speed	a speed and its unit of measurement
Condition	a condition
Method	a method
Note	notes in cases where action or descriptors are missing

Table 4.2: The information extraction template

4.3.3 Example of Processing SOP text

In this section, we describe the process of translating an example free text SOP with the SOP translator and demonstrate how it identifies all the essential information. We show how the SOP translator works using the urine microscopy procedure described in 4.2.2. We used the urine microscopy SOP provided by NICE SMIs [108] as an example of a free text SOP. To run the SOP translator, we carried out the following steps:

1. We pre-processed the urine microscopy SOP by removing all formatting and creating a plain text file in .txt format. Figure 4.3 shows a fragment of the SOP in natural language

2. We used the SOP translator to process the SOP

3. The SOP translator generated a csv file as the output file. Figure 4.7 shows a fragment of the output file.

Output of SOP processing

Statement	Action	Entity	Volume	Equipment	min Period	max Period	Condition	Note
Mix the urine gently to avoid foaming.	mix	urine::					gently::	No period specified for mix::
Using a pipette and disposable tips dispense known volume (~60µL see 'Note 2' below) of mixed urine to a numbered well in a flat-bottomed microtitre tray.	dispense	urine::	60µl::	pipette::well::flat-bottomed microtitre tray::				
Make sure that the specimen covers the whole bottom surface area (the use of a template will facilitate matching the specimen and well number).	make	specimen::		plate::well::				
Allow to settle for a minimum of 5min but preferably 10–15min before reading with an inverted microscope.	allow				5min::10 min	5min:15min		entity not found for action allow::No condition specified for allow::
reading with an inverted microscope	read			inverted microscope::				entity not found for action read::

Figure 4.7: Sample of output file from the SOP translator

The output file shows that the SOP translator identified a total of 16 actions from 10 sentences. The SOP translator identified some of the descriptors and where it could not find a required descriptor, it added a note. One problem we observed is that even though the entity specimen for the action allow was present in the preceding sentence, the SOP translator failed to identify it. The output file shows that the SOP translator was able to identify the action read the term reading in the last statement, which proves that the lemmatization technique works in the protocol tagger. There are additional information that start with the keyword note, so the SOP translator did not process those. However, two statements are also additional information but are missing any of the keyword note, caution, or warning so the SOP translator attempted to process the statements.

We mapped the content of the output file to OCL-SOP using Protégé tool as shown in figure 4.8. This also allows us to export the content of the output file into RDF and XML format. We processed several other SOPs with the SOP translator including the Malaria Microscopy test SOP. In section 4.4, we describe

how the SOP output files is used within the SmartSOP mobile application

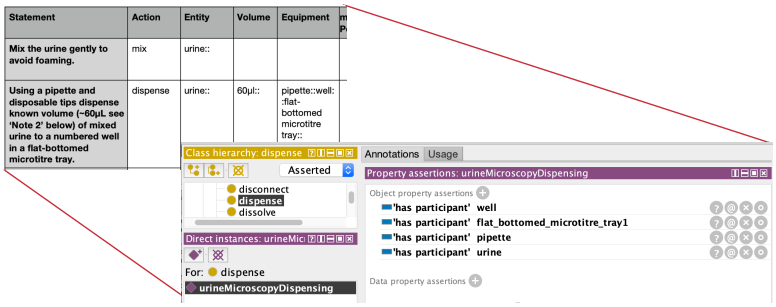

Figure 4.8: Urine Microscopy SOP mapped to OCL-SOP in Protégé

Expressiveness of the output file

The output of processing the Urine Microscopy SOP text gives an indication of the expressiveness of the machine readable version. Based on the description above, the output file shows the main components of the procedure but some details are missing. The output file does not show notes and tips given within the free text SOP, which are very helpful for the laboratory scientists. The advantage of the machine readable version over the free text version is that the information required to carry out actions are more complete (and where it is missing, a note indicates such). Whereas in the free text version, if any vital information is missing, it may not be clear. For example, some actions may require a temperature and this will vary by laboratory, if the temperature is not indicated in the free text SOP, the laboratory scientists may not be aware that it is important to use a particular value.

We can improve the expressiveness of the machine readable SOPs with further revisions of the OCL-SOP and the translation engine.

4.3.4 Evaluation of the SOP translator

In this section, we present the result of a lightweight evaluation we carried out on the translation engine to measure the performance of the tool. We tested the

tool on a small dataset of 10 SOP text that were not used in the development of the OCL-SOP or training the entity models for the NER. We measured the precision, recall and F-measure of the tool in NER of action entity. We carried out an analysis of errors from the testing. However, there is need for further evaluation of this tool and we describe this in the future work section of this thesis.

Evaluation metrics

We used the standard evaluation metrics of:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$F - measure = \frac{TruePositive}{TruePositives + FalsePositives}$$

Result

We processed the 10 selected SOP texts with the SOP translator. The SOP translator identified 153 occurrence of the action entities. Table 4.3 shows the top 10 most frequently used actions in the SOP texts.

Entity name	frequency
incubate	14
test	13
examine	11
place	11
filter	9
add	7
inoculate	6
slide	6
allow	5

Entity name	frequency
stick	4

Table 4.3: Top 10 named entities for action type

The performance measure of the tool in identifying the action entity occurrences is as follows:

$$Precision = 82\%$$

$$Recall = 94\%$$

$$F - measure = 88\%$$

Analysis of error

This evaluation measured the precision, recall, and f-measure for the NER of action entities. It will be beneficial to compare the performance of the SOP translator against other standard biomedical text processing tools such as MetaMap and cTAKES. This will be carried out as part of the future work of this research. During this testing, the following main errors occurred:

- The false positives were as a result of the tool wrongly identifying actions from tokens that have more than one POS tag. For example, the words *slide* and *filter* are both verb and noun. In addition, the trained action entity model contained the verb form of the words as actions. The tool could not differentiate when the words appeared as verbs from nouns.

- In a similar manner to above, some false positives are as a result of wrongly identifying part of a phrase, which represents other entities such as equipment, as actions. For example, the *test* part of *test tube* is identified as an action

- Another reason for a false positive is identification of all verbs as potential actions. In the protocol parser, if a token/lemma, which is tagged as a

verb in the POS tagging was not recognised based on the NER, then it assumes it is an action that is missing from the lexicon. Not all such cases are true as some of the verbs are not actions.

- There were a few false negative errors, which we could not determine a cause for. In those cases, the entities were captured in the trained model but the protocol parser still failed to identify them in the text.

4.4 The mobile application

The mobile application is the third component of SmartSOP framework. The main aim of the mobile application is to provide easy access to semantically annotated SOPs generated from the SOP translator. The mobile application also allows laboratory scientists to perform other functions such as recording test results and checking additional information on how to carry out procedures using their mobile devices. The current practices in the clinical laboratory requires the lab scientists to use free text SOPs which are either provided in soft-copy in PDF or MS Word format on a desktop computer or a hard-copy. In a few instances, the laboratory provides summarised versions of SOPs as flowcharts. The lab scientists find it challenging to find the required information about a lab procedure using the current practices [80]. In addition, to record the outcome of procedures, the lab scientists need to first write it down on a piece of paper and later transfer it to either a computer-based record system or a manual record-keeping system. This approach is problematic as it gives opportunities for human error, makes sharing the results outside the laboratory difficult, and does not guarantee the confidentiality of patient data. The mobile application addresses these challenges by giving easy access to the SOPs and facilitating the recording and sharing of test results. The lab scientists can record the results of a procedure at the time they are carrying it out. The mobile application stores the results in a universal format in a secured database, which makes it easier to share the results with authorised persons outside the lab.

The availability and use of mobile applications in the clinical laboratories is limited [103]. The use of mobile applications to support clinical laboratory procedures is efficient and cost-effective. Therefore, it is beneficial to encourage their adoption, especially in resource-scarce developing nations [80]. We have reviewed the existing clinical laboratory mobile applications such as e-learning application for pathology students [45], mobile microscopy application [27], and lab results reference applications [136, 61] in chapter 2.

We mentioned at the beginning of this chapter that we developed the mobile application over several iterations with the first version developed as a final year undergraduate project. The author of this thesis developed subsequent versions, including the final one described in this chapter. To build the mobile application component, we used the agile method of rapid prototyping technique. Rapid prototyping allows designers to use an iterative process of building early versions of their software to verify that they have satisfied all requirements [72]. Rapid prototyping technique provides a cost-effective and faster approach to building software unlike other methodologies such as the waterfall model which takes a considerably long time [84]. Figure 4.9 shows the stages in the rapid prototyping technique, which can be carried out iteratively until the final prototype meets all requirements.

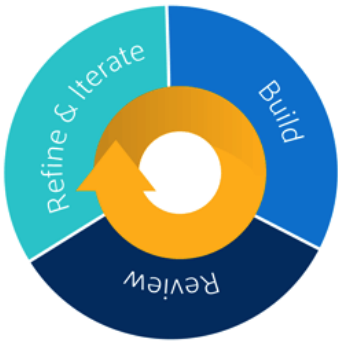

Figure 4.9: Rapid Prototyping

4.4.1 Development environment

The first stage is to develop the prototype, then review it, and finally refine and iterate. We used Balsamiq [12] and Lucidchart [77] as the prototyping tools to build mock-ups of the mobile application. These tools are available as web applications and offer the convenience of no installation. For each iteration, we discussed the prototype with the domain experts from the clinical laboratory to obtain feedback, which we used to refine the prototype for the next version. We carried out a total of three iterations and developed the mobile application based on the final prototype from the third iteration.

We built the mobile application in Apperyio [39] environment, which supports the languages and technologies we required to build the application user interface and connect to our data source. Figure 4.10 shows the software architecture of the mobile application. The mobile application is cross-platform and thus can work for both IOS and Android operating systems. The languages we used are HTML, CSS, and the jQuery JavaScript library to build the user interface. To use the system, the mobile client connecta to the webserver through the internet or LAN. The web server consists of our data store, which is a NoSQL database. NoSQL databases offer high availability and fault tolerance, which makes it suitable for applications that require multi-user access. The NoSQL database contains the machine-readable SOPs and machine-readable test results. The mobile client accesses the SOPs and sends the tests results to the NoSQL database by connecting through a REST API.

4.4.2 Defining high-level functionalities

The mobile application is the outcome of RO4, which specifically aims to deliver a mobile application for clinical laboratory scientists to have easy access to SOPs and monitor their adherence to guidelines. We identified two of the high-level functionalities of the mobile application based on RO4; access to SOPs and keeping track of who uses the SOPs. Through our refinement of proto-

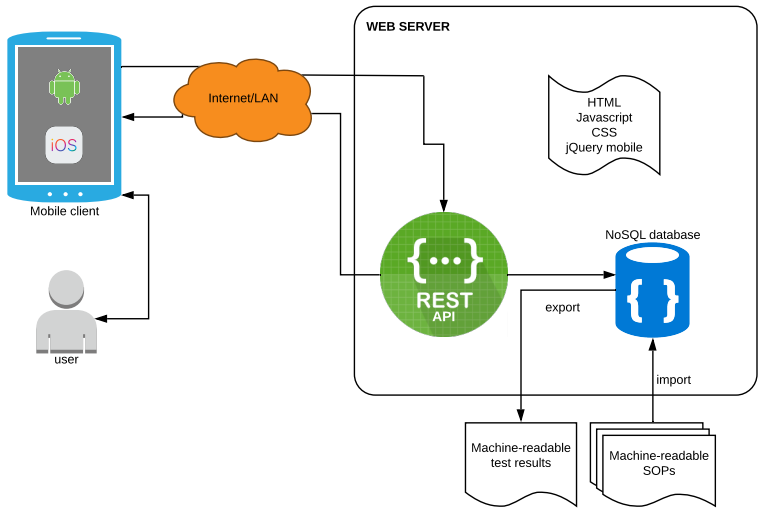

Figure 4.10: SmartSOP mobile application software architecture

type based on feedback from the domain experts and our observations at the clinical laboratories, we identified another high-level functionality, which is to record the test results on the application. In figure 4.11, we describe the features that satisfy the high-level functionalities in the mobile application. In

High level functionality	Detailed feature/description	OCL-SOP class
Login to mobile application	• Creating usernames and passwords • Logging in to the app to keep track of the SOPs each user accesses, the procedures they carried out, and the test results they recorded.	Not applicable (future addition to OCL-SOP)
Access SOPs	• Checklist of resources • Steps for carrying out procedures • Videos for complicated steps • Search feature	• Equipment and biochemical entity (or material entity) • Experimental actions corresponding descriptors
Record test results	• Automatically manipulate data elements • Record results • Export results	• Data actions • Laboratory findings

Figure 4.11: High-level functionalities of SmartSOP mobile application

the previous section, we described the output of the SOP translator, which is in machine-readable format. We developed the mobile application to read from the description of the SOP from the output file. The ontological representations in OCL-SOP allowed us to understand the characteristics of the SOPs that we need to show in the mobile application. In figure 4.11, we present the features of the mobile application and their corresponding representation in OCL-SOP. Also, we identified that each SOP should begin with a list of resources the lab scientist needs to carry out a procedure. We also identified that the recording of the test results occurs at the end of the procedure. This information allowed us to design the user interface so that it represents the actual flow of procedures, starting with test selection, then a checklist of resources, the actual procedure steps, and finally recording the test results.

4.4.3 Features of the mobile application

The mobile application has features, which enable the following core functionalities. Figure 4.12 shows the use case diagram for this application.

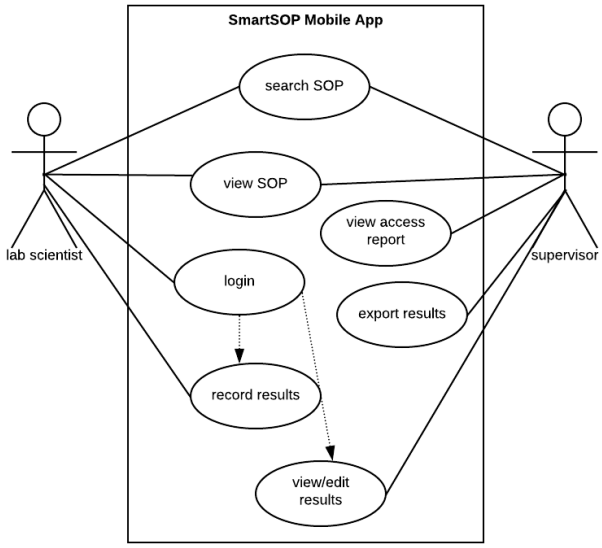

Figure 4.12: Use case diagram for the SmartSOP mobile application

Login The mobile application provides a secure login feature to ensure that only approved personnel in the clinical laboratory can access and use the SOPs. Individual clinical laboratories are responsible for creating access for their users. The application keeps track of the activities of the users, which is essential for auditing and monitoring adherence to guidelines. The first step in monitoring adherence, is having a record of who did what and when on the system. Restricting access to the mobile application to only authorised users is also vital for viewing and exporting test results, which is important to ensure patient data remains confidential.

Access to SOPs The core functionality of the mobile application is to provide clinical laboratory scientists with easy access to SOPs. We designed the application to display a list of available test procedures on the first screen after login. We had several discussions with the domain experts to determine the best approach for categorizing the SOPs on the mobile application. The experts recommended using a standard approach of grouping the SOPs based on the kind of tests, for example, malaria test can contain several procedures such as malaria microscopy test, rapid diagnostic test, with each SOP describing how to carry out one procedure. Figure 4.13 shows an example of the categorization of the SOPs for the malaria test and how to navigate to a specific test, in this case the malaria microscopy test. For each SOPs, the mobile app dis-

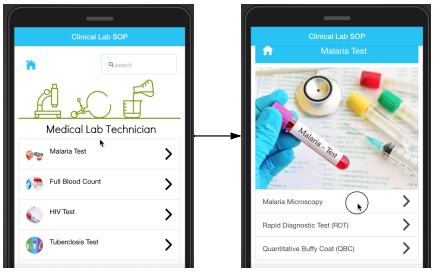

Figure 4.13: SOPs in SmartSOP mobile application

plays a list of required resources for carrying out the procedure. We provided this function as a checklist which the lab scientists need to click through to

verify that all essential equipment and biochemical entities are available. If the checklist is not complete, the application prevents the user from continuing to the next functions, and as a result, they cannot continue with the procedure. This check is necessary to ensure quality of the test procedure by reminding lab scientists of the materials they need and also preventing them from carrying out the test without these materials. Figure 4.14 shows the checklist for the malaria microscopy test procedure. The mobile application can provide additional information for the materials by either showing a definition or an image of some key meterial when the user clicks on the link. For example, the mobile app can show a description of how to make the Giemsa stain, a material required for the malaria microscopy test. After successful verification of the checklist, the

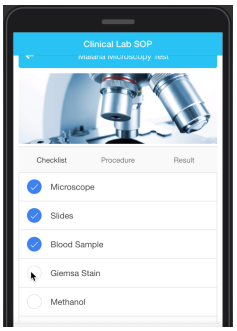

Figure 4.14: Checklist of materials in SmartSOP mobile application

mobile app shows the steps required for the procedure. The steps are sentences that describe the procedure, which are extracted from the machine-readable SOPs. One of the components of the output file from the SOP translator is the Statement which contains the sentences from the SOP. Each step in a procedure provides all the necessary information to carry out the task. The mobile app provides additional information such as images or definition of some key terms. In figure 4.15, we show a fragment of the steps for the malaria microscopy test SOP. We also included videos for some steps that are complicated and may be difficult to follow from written instructions. An example of a video, showing how to prepare a 'thick smear' is seen in figure 4.15. Another important feature that is part of the access functionality is the search feature. The search

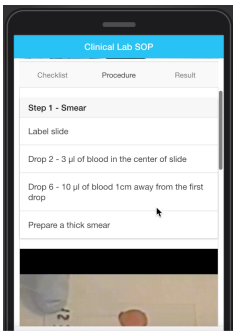

Figure 4.15: Fragment of malaria microscopy procedure

feature allows users to find available SOPs which contain the keyword used in the search. If the search keyword is an experimental or data action, the results include a general description of how to carry out that action. This is useful in a case where the user only wants to find out how to do that action without necessarily carrying out a procedure from any particular SOP. For example, if a user wants to find information on smear which is an experimental action, they search with the keyword smear and among the results there is a description of how to carry out the action along with a video tutorial

Recording test results There are some procedures that require calculations to derive some data element that are recorded as the test result or it is used to further calculate the result. The mobile application automates these calculations for the relevant SOP, which are carried out in the results entry section of the mobile application. Each SOP contains a results entry section where the lab scientists can record one or more test results. The results section consists of input fields for collecting data values from the lab scientists, buttons for calculations where they are required, and submission button to save the test results into the database. Figure 4.16 shows the results entry section for the malaria microscopy test SOP. The results entry section differs for each SOPs as the different tests observe, calculate and record different data elements. We designed and built the results entry section to fit each specific SOP. The results are recorded in a secured NoSQL database, this allows us to later retrieve and view

the results and also export the results for sharing outside the clinical laboratory. The recording results functionality provides an improvement over the manual process of recording and sharing test results. It reduces the error rate that is associated with a paper-based record system and also enable interoperability with systems outside the laboratory. The results in the NoSQL database are in a machine-readable format which we can easily map to the classes in OCL-SOP. 4.15.

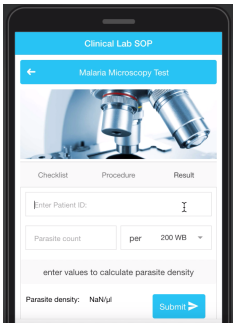

Figure 4.16: Results screen for malaria microscopy procedure

4.5 Summary

In this chapter, we described the different components of the SmartSOP framework and how it provides tools that support the representation and use of SOPs in clinical laboratories. The SmartSOP framework forms our main technological contribution in this thesis. The framework consists of three components, the OCL-SOP, SOP translator, and the mobile application. We have previously described the development and structure of the OCL-SOP in chapter 3. In this chapter, we explained how the OCL-SOP provides a data model for the SmartSOP framework. We discussed the SOP translator and the main changes we applied to the previous version. We explained how the translator converts free-text SOPs into machine-readable format. Finally, we described the mobile application and how it uses the output of the SOP translator and enables recording of test results.

We carried out a user-centred evaluation of the SmartSOP framework, which we discuss in chapter 6. In chapter 5, we introduce external projects that have used the SmartSOP framework.

Chapter 5

SmartSOP Framework in Practice

In this chapter, we describe research collaborations, which led to the use of SmartSOP framework within projects outside this research work. This chapter demonstrates the usefulness of the SmartSOP framework and added value to the research community. There are currently three projects that have successfully utilised our proposed framework, which are:

- Development of the Neurodegenerative Disease Data Ontology
- IEEE Robot Task Representation working group
- Maholo LabDroid

In the next sections, we present a brief description of each project, our contributions or involvements with the projects, and how the SmartSOP framework is featured in the research works. Parts of the work we describe in this chapter are published in [73] and [143].

5.1 The Neurodegenerative Disease Data Ontology

The Neurodegenerative Disease Data Ontology (NDDO) is an ontology that represents brain diseases data [73]. The developers created NDDO in response to the needs of the Human Brain Project (HBP) to develop a wide range of ontologies for brain diseases and types of data [65]. The HBP is part of the Future and Emerging Technologies Flagships, which are the largest funded projects in the European Union. The HBP aims to create a strong research base for advanced neuroscience, medical, and computing researches [66]. There are numerous projects currently running under the ICT research platforms of the HBP across several universities, teaching hospitals, and research centres in Europe. These ICT research platforms are Neuroinformatics, Brain Simulation, High-Performance Analytics and Computing, Medical Informatics, Neuromorphic Computing, and Neurorobotics.

The HBP explore the complexity of the brain to understand how it functions and diseases that affect the brain and create interdisciplinary research in developing health, computing and technology applications [66, 5]. As such, the call for expression of interest on comprehensive ontologies for brain diseases, which the NDDO is in response to, required that the ontologies should be interoperable with existing ontologies and hospital data.

NDDO is consistent with the structure of hospital data from two neurodegenerative diseases studies; the Alzheimers Disease Neuroimaging Initiative (ADNI) and Parkinsons Progression Markers Initiative (PPMI) [73]. NDDO is also compatible with existing standards for brain diseases and is easily extensible to incorporate related ontologies to represent brain diseases data fully. NDDO provides a formal representation of data collected in ADNI and PPMI which consists of entities such as the study participants, their visits of physicians, different assessments conducted during that visits and their results, and the diagnosis. [73]

ADNI and PPMI follow a set of standard procedures to generate the data represented by NDDO; for example, PPMI uses a laboratory procedure protocol for the analysis of haemoglobin levels in CSF samples. NDDO does not capture these protocols that generate the disease data. Therefore there is a need to align it with a suitable ontology. In this thesis, we have presented OCL-SOP, which provides a formal representation of clinical laboratory SOPs. OCL-SOP adequately represents terminologies within ADNI and PPMI laboratory procedure protocols but does not represent the description of disease data, thus offering a suitable ontology for alignment with NDDO.

Also, like most clinical laboratory standards, the ADNI and PPMI laboratory procedure protocols exist in natural language. We have argued that representing protocols in natural language creates several challenges such as lack of standardisation, which can lead to different interpretations by different agents, and consequently to different implementations and outcomes. Moreover, it complicates their computational processing and analysis, e.g. it is difficult to compare procedures expressed in natural language, to identify gaps, and to check them for logical consistency and completeness. [73]

In this section, we describe the alignment of OCL-SOP to NDDO to create a network of ontologies that adequately represents clinical procedures used to produce neurodegenerative data and the data. We also describe how the Smart-SOP framework processed an example of a laboratory procedure protocols from PPMI.

5.1.1 Aligning OCL-SOP with NDDO

The structure of NDDO includes a continuant entity laboratory finding which is in the hierarchy of 'data item'–¿'clinical finding'–¿'laboratory finding'. NDDO, like OCL-SOP, adopts the classifications of continuant entities and processes from BFO, which makes it easy to align the two ontologies. To align OCL-SOP with NDDO, we identified and imported the relevant terms 'data item'–¿'clinical finding'–¿'laboratory finding' from NDDO into OCL-SOP. Figure 5.1 shows the

imported classes in OCL-SOP. OCL-SOP contains a data action branch which

Figure 5.1: Classes imported from NDDO into OCL-SOP

represents actions found within the clinical laboratory protocols that generate data items. We related the imported class laboratory finding with the data action branch using the has_specified_output object property, as shown in figure 5.2.

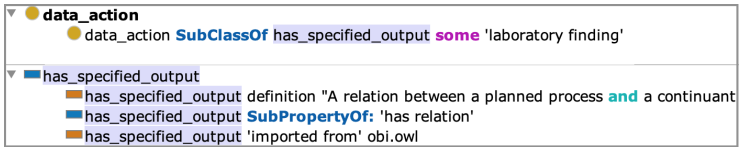

Figure 5.2: Showing the relation between 'data action' and 'laboratory finding' [73]

5.1.2 Processing PPMI Protocols with SmartSOP framework

The alignment of OCL-SOP with NDDO allows the SmartSOP framework to process ADNI and PPMI laboratory procedures protocols to annotate with OCL-SOP terms, generate machine-readable versions, and use the protocols within the mobile application. To show an example of how to translate protocols from this project with the translation engine, we processed the protocol for testing for the presence and levels of hemoglobin in cerebrospinal fluid (CSF) [73]. The PPMI analysis of haemoglobin in CSF samples procedure requires the use of an ELISA assay obtained from Bethyl Laboratories [4]. Figure 5.3 shows a segment of the Human Haemoglobin Elisa Kit (HHEK) protocol in natural language. We updated the list of equipment and chemical entities for the translation engine with new materials found in the HHEK protocol. We then

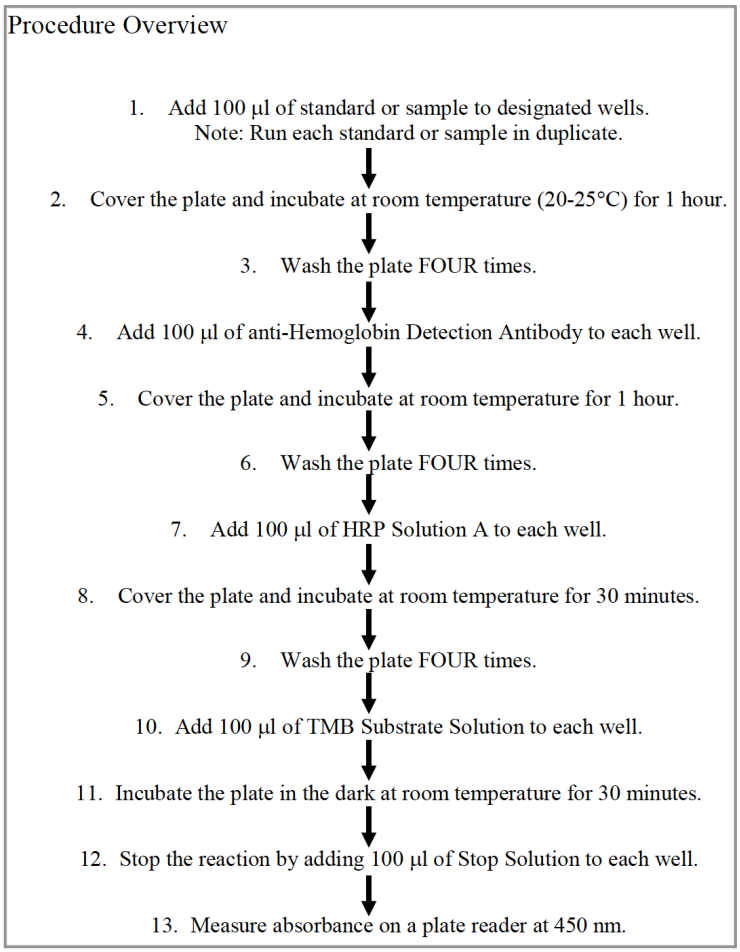

Figure 5.3: Human Hemoglobin Elisa Kit Procedure Overview [73]

processed the protocol and converted it to machine-readable format. Figure 5.4 shows the content of the output file from the translation engine.

Statement	Action	Entity	min Temperature	max Temperature	Volume	Equipment	Period	Condition
Add 100 µl of standard or sample to designated wells.	add				100 µl::	wells::		
Note: Run each standard or sample in duplicate.								
Cover the plate and incubate at room temperature (20-25°C) for 1 hour.	cover					plate::		
incubate at room temperature (20-25°C) for 1 hour.	incubate		18°C:UO:0000027	18°C:UO:0000027			1 hour	at room temperature::
Wash the plate FOUR times.	wash							
Add 100 µl of anti-Hemoglobin Detection Antibody to each well.	add				100 µl::	well::		
Cover the plate and incubate at room temperature for 1 hour.	cover					plate::		
incubate at room temperature for 1 hour.	incubate		18°C:UO:0000027	18°C:UO:0000027			1 hour	at room temperature::
Wash the plate FOUR times.	wash							
Add 100 µl of HRP Solution A to each well.	add	HRP Solution A::			100 µl::	well::		
Cover the plate and incubate at room temperature for 30 minutes.	cover					plate::		
incubate at room temperature for 30 minutes.	incubate		18°C:UO:0000027	18°C:UO:0000027			30 minutes	at room temperature::
Wash the plate FOUR times.	wash							
Add 100 µl of TMB Substrate Solution to each well.	add	TMB Substrate Solution::			100 µl::	well::		
Incubate the plate in the dark at room temperature for 30 minutes.	incubate		18°C:UO:0000027	18°C:UO:0000027		plate::	30 minutes	in the dark::at room temperature
Stop the reaction by adding 100 µl of Stop Solution to each well.								
Measure absorbance on a plate reader at 450 nm.	measure							

Figure 5.4: Segment of Output File content from the Translation Engine [73]

5.1.3 Uses of the machine-readable protocols

In this project, we report that having rich semantic representations of neurodegenerative data will enable more efficient data mining and knowledge discovery. For example, information about the procedures should be used for the integration of datasets to ensure that all the analysed data were collected following similar procedures. [73]

The output file for the processed PPMI protocol is machine-readable and processable, which makes it available for a wide variety of computational use. For example, the output file includes a note to show where the protocol is missing key entities, and we can use this information to update the protocol, which will improve the quality of the procedure. Another example is that we can use the Smart SOP mobile application to read the output file, display it to laboratory practitioners, and allow them to record the outcome of the measure data action as a semantically annotated laboratory finding value.

5.2 IEEE Robot Task Representation working group

The IEEE Robot Task Representation working group is responsible for developing a robot task ontology for knowledge representation and reasoning in robotics and automation. Researchers from academia, industry, and government from across the world constitute the members of this working group. The author of this thesis is an active member of the working group and is directly involved in the development of the robot task ontology. The author is responsible for describing an example implementation of the ontology. At the time of authoring this thesis, the working group have not completed the ontology development. Although the group have completed, significant portions of the work, we are still careful of the amount of information we can present in this section. Therefore, we only provided a brief description of the example implementation. The exam-

ple implementation we describe in this section is currently in a manuscript, in which the working group prepared to give a preliminary report of the groups.

5.2.1 Aligning OCL-SOP with the robot task ontology

The robot task ontology will provide an upper-level ontology for domain-specific and application ontologies to extend, in order to represent a description of robotic tasks. As an example of such implementation, we considered OCL-SOP as a domain ontology that can extend the robot task ontology to describe tasks that robots can accomplish using clinical laboratory SOPs.

The structure of the proposed robot task ontology consists of an actions and sequence frame, which provides a structure for describing the actions and relevant parameters that a robot will need to carry out to accomplish a task. We will extend the action and sequence frame through the processes class in OCL-SOP to describe experimental and data actions as subclasses within this frame. This will allow a description of the actions a clinical laboratory robot will need to carry out within the lab. In the next section, we illustrate an example of a clinical laboratory robot that will carry out malaria microscopy test in the lab and how the robot task ontology action and sequence frame provide a framework for description of the relevant OCL-SOP actions.

5.2.2 Automation of the malaria microscopy test procedure

In recent years, laboratory automation systems are rapidly evolving to provide support for complex processes in the lab aimed at supporting clinical diagnosis. The advantages of laboratory automation include standardisation of the testing processes, increased accuracy in test results, reduced turnaround time, reduced risk of laboratory-acquired infection for the personnel, and substantial cost savings [26]. Robotics in the clinical laboratory is useful for automation of both analytical and non-analytical processes. One example of robotics application in

the clinical laboratory is the Mobile Agent, which transfers materials around the lab (non-analytical) and performs various testing procedures (analytical) [32]. In the clinical laboratory, some procedures are similar and have differences in only the specimens analysed and the expected outcome of the analysis. In such cases, it is beneficial and cost-effective to have laboratory robots that can be reconfigured to carry out different procedures. It is essential to standardise the representation of the robot tasks to develop multi-functioning robots in the lab, which is one of the goals of the robot task ontology.

We propose an automation of the malaria microscopy test, which is the gold standard for identifying and specifying malaria parasite from blood samples in the clinical laboratories. We can achieve the automation of the malaria test using laboratory robots. The proposed robot can also be reconfigured to carry out other microscopy test such as urine microscopy to identify the presence of cellular components. There is an alarming shortage of malaria microscopy test expertise, especially in malaria-endemic countries such as Nigeria where the national malaria prevalence is 23% [92]. Without adequate expertise, the malaria test is susceptible to problems that will lead to inaccuracies in testing and interpretation of the results [97]. The malaria test robot we propose will reduce such challenges by improving the reliability of the test.

Figure 5.5 shows a representation of the actions which the robot needs to complete in order to carry out the malaria microscopy test. Once the working group completes the robot task ontology, we can describe the malaria microscopy robot task more holistically.

5.3 Maholo LabDroid

A team of researchers at the University of Tokyo and Robotic Biology Consortium have proposed a laboratory humanoid robot, Maholo LabDroid. The Maholo LabDroid is introduced as a new research concept of robotic crowd biology, which automates labour-intensive and bio-hazardous laboratory procedures

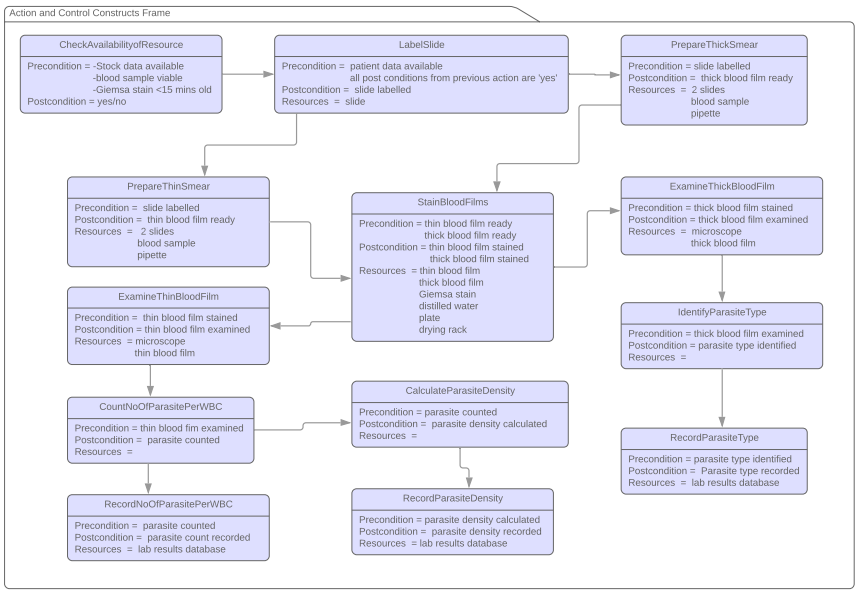

Figure 5.5: Malaria microscopy test automation actions

to make them safer and more efficient [143]. Like the proposed malaria test robot, the Maholo LabDroid is based on ontological descriptions of laboratory procedures.

The Maholo LabDroid also provides opportunities for collaboration between humans and robots while carrying out laboratory procedures. The LabDroid executes the laboratory experimental actions using similar techniques as humans. Although [85] has criticised this factor by arguing that more advanced systems such as liquid handling and integrated instrument technologies already exist, the LabDroid has the unique advantage of being able to work alongside human scientists in the lab.

5.3.1 Laboratory actions

We contributed to the development of the Maholo LabDroid by analysing several clinical laboratory procedures to identify the most frequently used experimental actions for inclusion in the robot. Maholo LabDroid used the ontology EXACT as its ontological model for describing the experimental actions. The Maholo

LabDroid has a graphical user interface, which is based on the descriptions from the ontology EXACT that is used to configure actions the robot is required to carry out for experimental procedures. We identified several experimental actions from the laboratory procedures and mapped the actions to the ontology EXACT during our analysis. Figure 5.6 shows a fragment of the results of the analysis. Figure 5.7 shows an image of the Maholo LabDroid.

List of Frequently Used Experimental Actions in NICE SMIs for Test

Experimental Action	Usage*	Procedures**	Synonym in EXACT2
incubate	9	6	
add	9	6	
examine	6	6	
place	4	3	move
mix	4	4	
allow	4	3	
discard	3	3	
inoculate	2	2	
divide	2	2	
dilute	2	1	

Usage* - total number of times the action appeared in 10 randomly selected procedures
Procedures** - total number of procedures the action appeared in.

Figure 5.6: Experimental actions in Maholo LabDroid

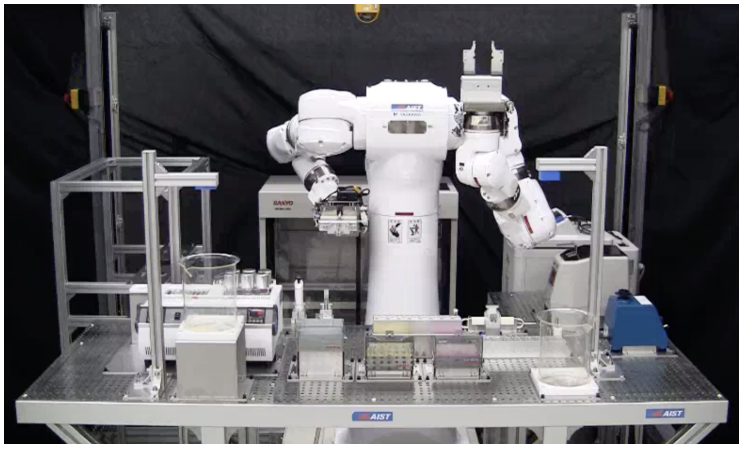

Figure 5.7: Image of the Maholo LabDroid

Chapter 6

Evaluation of the SmartSOP Framework

This chapter aims to describe the evaluation of the main contribution of this research, the SmartSOP framework, and present the result of this evaluation. We carried out the evaluation to assess if the proposed framework enables an accurate representation of terminologies from clinical laboratory SOPs, provides easy access to the SOPs and also encourage adherence to the SOPs. This chapter addresses our research objective 5, which is to evaluate the effectiveness and user satisfaction of the proposed framework in this thesis.

In design science, the main goal of the evaluation activity is to assess the extent to which an artefact effectively solves the problem for which it was proposed [60]. In chapter 1, we mentioned that design science has the flexibility of allowing the researcher to focus on some research activities while treating other activities lightly. In this research, we carried out a lightweight evaluation activity to determine how well the SmartSOP framework alleviates the problems that motivated our work. Design science also prescribes the evaluation activity for other purposes, such as, to evaluate the requirements of the artefact; to investigate a formalised knowledge about an artefact; to compare a proposed

artefact to other artefacts; to investigate the side effect of using an artefact, or to carry out a formative or summative evaluation [69]. A formative evaluation is carried out while the artefact is still being developed to inform the developer on areas for improvement. In chapters 3 and 4, we described some formative evaluation activities that informed improvements in our requirement specifications for the different components of the SmartSOP framework. While adopting design science, researchers can carry out an evaluation based on one or more of the purposes we have previously mentioned. We also decided to carry out a summative evaluation, which is typically used to obtain a final assessment the artefact at the end of the development [69]. This summative evaluation allowed us to measure the usability of the SmartSOP framework. [20] states that the findings of summative evaluations are meant to confirm and validate the usability of a proposed system, which is what we aim to achieve with this evaluation.

The rest of the chapter is organised as follows. In section 6.1 we provide the evaluation objectives, the evaluation approach we adopted is covered in section 6.2, section 6.3 describes the experiment we carried out, and we present the results from the experiment in section 6.4.

6.1 Evaluation objectives

The main objective of this chapter is to address our research objective 5, which is to determine the effectiveness and user satisfaction of the SmartSOP framework in representing the clinical laboratory SOPs and providing easy access to the SOPs. To address this objective, we divided it into the following specific objectives:

1. To determine the correctness and completeness of information from the SOPs as represented by the SmartSOP

2. To assess the ease of access to information on the SmartSOP mobile application and if it will encourage adherence

3. To measure the user satisfaction for the SmartSOP framework

6.2 Evaluation approach

We designed the evaluation using a mix of research methods as recommended for evaluating artefacts in design science. To address the specific evaluation objectives from the previous section, we carried out a usability evaluation through an experiment in real clinical laboratories and collected data through observations and a self-administered questionnaire. The data we observed and collected through the questionnaire measures metrics for either the effectiveness (specific evaluation objectives 1 and 2) or the user satisfaction (specific evaluation objective 3).

[18] describe the usability evaluation as a fundamental step for ensuring that systems are adapted to the users, their tasks, and that there are no adverse outcomes of their usage. The usability evaluation measures how well a system carries out the tasks it is designed for (effectiveness), the amount of resources required to use the system (efficiency), and how well the users respond to the system (user-satisfaction) [98].

There are three primary methods for carrying out a usability evaluation, the inspection-based, user-based, and model-based evaluations [18]. The inspection-based evaluation can occur at any stage of the development and is carried out by a trained analyst who examines the proposed system or a prototype [118]. The model-based evaluation are mainly used for large research projects and have high associated costs [117]. The unique characteristic of the user-based evaluation is that it involves the intended users of the proposed system carrying out a task while their behaviours are observed and recorded. Usability practitioners widely use the inspection-based and user-based evaluation, which are well documented, whereas the model-based evaluation is not as popular [18]. In our usability evaluation, we employed the user-based evaluation approach, where the clinical laboratory scientists (intended users) used the proposed framework

in an experiment while we observed and recorded their behaviour and responses to the system.

6.3 Experiment

To carry out the usability evaluation, we designed and conducted an experiment in clinical laboratories. In this section, we describe details of the experimental setting, the participants, the experimental procedure, the data collection, and data analysis methods. In section 6.4, we discussed the results of this experiment.

6.3.1 Experimental setting

In chapter 1, we described the background of this research and highlighted some of the challenges that informed our research aim are lack of standardisation and adherence to SOPs. Clinical laboratories face these challenges worldwide, but they are more prevalent in developing nations. The WHO has set strategic frameworks and guidelines to strengthen clinical laboratories in developing nations, this includes mandating the use of quality SOPs. Throughout this research, we worked with clinical laboratory experts from the UK and in Nigeria to identify requirements and get feedback during the development of the SmartSOP framework. In order to demonstrate the usefulness of the SmartSOP framework in addressing the challenges in the developing countries, we chose Nigeria as our country of evaluation.

We carried out this evaluation in two cities in Nigeria, Abuja and Zaria. We identified three clinical laboratories, one in Zaria and two in Abuja. The laboratory in Zaria is based in a government-owned teaching hospital, which is one of the most prominent clinical laboratories in the country. The laboratory in Zaria is also a reference lab in the country. In Abuja, we selected one clinical laboratory that is based in a private hospital. Most hospitals in the country have an in-house clinical laboratory. However, depending on the size of the hospital,

the laboratory may not be fully equipped to carry out all kinds of test. This led to the creation of several independent clinical laboratories where patients from hospitals with smaller labs or with none at all are referred for testing. The third laboratory we selected is a private independent clinical laboratory in Abuja. The independent laboratory carries out tests on referred patients from hospitals and give the patients their results, which they take back to their doctors.

Our selection of clinical laboratories allows us to have a fair representation of the different kinds of hospitals that exist in the country. With different laboratories, we are also able to evaluate specific functionalities of the SmartSOP framework such as processing lab-specific SOPs and sharing test results outside the laboratory.

6.3.2 Participants

The participants consisted of 17 laboratory scientists from the three clinical laboratories. Six lab scientists were from the teaching hospital laboratory, six were from the private hospital-based laboratory, and five from the independent laboratory. The only requirements for the participants will be that they own a smartphone and have experience with using mobile applications.

To characterize the participants, we used the questionnaire (see section 6.3.4.) to collect personal attribute data of age and years of experience working in the laboratory. To understand the level of engagement with SOPs, we collected data on how often the participants use SOPs in a week. The distribution of the age of the 17 participants is, 65% are between 18 29 years, 25% are between 30 39 years, 5% are between 40 49 years, 5% are 50 years or older. The distribution for the years of experience in a clinical laboratory is, 71% have worked for between 0 5 years, 17% between 5 10 years, and 12% for more than 10 years. Regarding the level of engagement with the SOPs per week, only 6% used it 4 5 times, 18% used it once, 18% never, 29% used it 2 3 times, and 29% used it more than 5 times. We found that the participants with more than

Participant characteristic	*N = 17*
Age	
18 – 29 years	65%
30 – 39 years	25%
40 – 49 years	5%
50 years and Older	5%
Years of experience	
0 – 5 years	71%
5 – 10 years	17%
More than 10 years	12%
Frequency of SOP use	
Never	18%
Once a week	18%
2 – 3 times a week	29%
4 – 5 times a week	6%
More than 5 times	29%

Figure 6.1: Summary of the characteristics of participants

10 years of experience use SOPs significantly less than those with fewer years of experience. During our discussions with domain experts, this issue was raised several times, and the consensus is that the veteran lab scientists are more comfortable performing the procedures and as a result feel have less need for SOPs. Figure 6.1 shows a summary of the characteristics of the participants.

6.3.3 Experiment Task

Before the start of the experimental task, we processed a chosen SOP with the translation engine and loaded the output file into the SmartSOP mobile application. We chose the Malaria Microscopy Test SOP and asked each of the three clinical laboratories to provide us with a PDF version of their lab-specific SOP. We provided and prepared a mobile device by loading the mobile application and the post-task questionnaire. We then conducted a short tutorial to explain the various features of the application.

The task required each participant to carry out one malaria microscopy test while navigating through the SOP on the mobile application. The laboratories freely provided the materials required to carry out the malaria microscopy test. The participant will then record the results of their observation on the mobile

application and export the test result using the built-in functionality in the mobile application.

6.3.4 Data Collection

At the end of the experiment, we provided a self-administered questionnaire to collect data and opinions about the usability of the mobile application. We designed the questionnaire to collect data that address the specific evaluation objectives of accuracy and completeness of information, ease of access, encouraging adherence, and user satisfaction. The questionnaire consisted of two parts, part one collected the participant characterization data (see section 6.3.2.), and part two asked the usability evaluation questions. We used the usability principles for measuring effectiveness and user satisfaction, recommended by [20], as a guide to form the questions in part two. Part two consists of 13 usability questions which are answered using a 5-point Likert scale. We also included one free text question in part two for the participants to add any comments or suggestions. We used Google Surveys to create the questionnaire and accessed the questionnaire through the mobile devices we provided for the experiment. The complete questionnaire is attached in appendix A.

6.3.5 Data Analysis

We analysed the data we collected through the questionnaire to assess the usability of the SmartSOP framework. We carried out a descriptive data analysis and present the results in section 6.4.

6.3.6 Ethical considerations

We presented the participants with an information sheet that describes the research and the purpose of the study. We obtained informed consent from each participant for taking part in the study and for us to use their responses in this thesis. The informed consent specified that they understand the study, they

QUESTION NO.	% NEGATIVE RESPONSE	% NEUTRAL RESPONSE	% POSITIVE RESPONSE
ACCURACY			
Q8	6%	18%	76%
Q11	0%	18%	82%
COMPLETENESS			
Q9	6%	29%	65%
Q10	6%	41%	53%
EASE OF ACCESS			
Q5	12%	6%	82%
Q6	6%	18%	76%
Q7	18%	18%	65%
ADHERENCE			
Q16	12%	29%	59%
Q17	0%	18%	82%
SATISFACTION			
Q12	18%	6%	76%
Q13	18%	18%	65%
Q14	6%	0%	94%
Q15	12%	29%	59%

Figure 6.2: Summary of responses from participants

are voluntarily participating, and they understand that their responses will be anonymous. We obtained permissions from the management of the three labs to carry out the study using the facilities.

We also obtained ethical clearance at the early stage of this research from the authors university at the time and the Health Research Ethics Committee in Nigeria.

6.4 Results

The data we collected through the questionnaire measures metrics that show if the framework is effective in representing correct and complete information from SOPs, makes it easier to access the information, encourages adherence to SOPs, and measure the user satisfaction. The effectiveness measure addresses the first and second specific evaluation objectives from section 6.1., and the user satisfaction measure addresses the third evaluation objective. Figure 6.2 shows a summary of the responses for all the usability questions. Majority of the participants responded positively to all the questions.

6.4.1 Effectiveness of the framework

The overall effectiveness is comprised of the average scores for the effectiveness metrics of accuracy of information, completeness of information, ease of access to information, and encouraging adherence to SOPs. We used a weighted scoring method to calculate the scores for the different effectiveness metrics. To determine the scores, we first assigned 1 point to each level on the Likert scale. So, if a response from the questionnaire is 1 on the Likert scale, then the point is 1, and if it is 4, the point is 4, and so on. Since we had 17 participants, the maximum points for each question is 85. Therefore, we calculated the score for each question as:

$$S = \frac{\sum_{i=1}^{17} p_i}{85}$$

where:

S = score for question

p_i = question response point for each participant

The various effectiveness metrics entail a set of questions from the questionnaire; for example, we measured the accuracy of information through responses from questions 8 and 11. Therefore, we determined the score for each metric as the average score for all the questions in that category. Figure 6.3 shows the result of the effectiveness measure.

The results of this analysis indicate that the overall effectiveness of the application is positive in achieving all its functionalities. The analysis of the individual metrics performance from this evaluation, show that the accuracy of information is 84%, completeness of information is 78%, ease of access to information is 81%, and encouraging adherence to SOP is also 81%.

6.4.2 User satisfaction

We measured the user satisfaction through a set of four questions from the questionnaire. We used the same weighted scoring method from the effectiveness

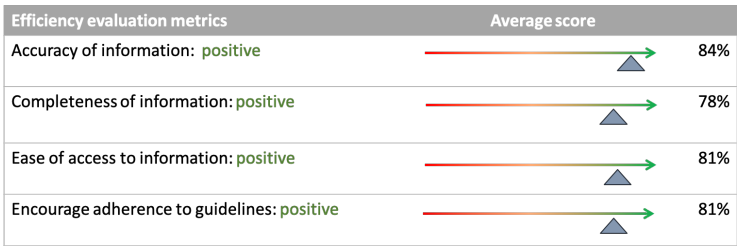

Figure 6.3: The overall effectiveness measure

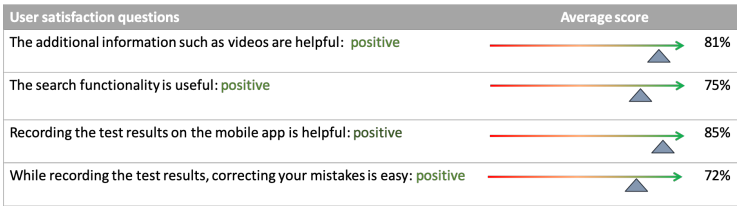

Figure 6.4: The user satisfaction measure

measure to determine scores for the user satisfaction questions. Figure 6.4 shows the result of the overall user satisfaction measure as the average scores assigned to the relevant questions. The results indicate a positive outcome for the user satisfaction evaluation with all the questions scoring above 70%.

The free text answers from the last question in the questionnaire also provide a measure of user satisfaction. There were only three responses that answered the free text question, and two were positive response that indicated the users were satisfied with the tool. The two responses are Very helpful and user friendly app! We need this in our labs and Its a pleasure having this new development. Wishing more of its kind will come up.

Chapter 7

Conclusion and further work

This chapter summarises the research work we presented in this thesis. We reflect on the research question and relate the different research contributions to their specific research objectives. We also describe further research work.

7.1 Summary

The main research question we addressed was how to standardise the representation of the clinical laboratory SOP and support their use in the laboratory. We identified several problems regarding the representation and use of SOPs in the clinical laboratory and proposed an IT-based solution. To ensure the quality of services in the clinical laboratory, it is crucial for the information in SOPs to be accurate and to use the SOPs as a guide to carry out lab procedures. The aim of this research was to provide a semantic-driven framework that improves knowledge representation and use of SOPs in clinical laboratories. Our primary research contribution is the SmartSOP framework that provides a solution to the practical problems of standardising SOP representation and improving usage in clinical laboratories. The SmartSOP framework consists of three components,

which are the OCL-SOP, translation engine, and the mobile application. Each of the SmartSOP component addresses at least one research objective. In this thesis, we have described the development and characteristics of the proposed framework. We also evaluated the framework to demonstrate how it addresses the research objectives. We described external research projects that have used or are currently using some of the outputs of this research work.

We started by reviewing the current literature to gain an understanding of the theoretical background we need for this research, which we presented in chapter 2. This addressed the research objective 1 by providing us with in-depth knowledge of how ontology is used for knowledge representation. The literature review also covered the OWL-DL language and the approach of reusing existing ontologies, including basing biomedical ontologies on the BFO to promote interoperability.

Based on the knowledge gathered from addressing objective 1 and following the basic approach of ontology reuse, we define an ontological model for the formal representation of clinical laboratory SOP terminologies (OCL-SOP). This addressed research objective 2, and in chapter 3, we presented the activities we carried out for the development of OCL-SOP as well as the structure of the ontology. We identified that currently, SOPs are represented using non-standardized terminologies based on the preferences of individual clinical laboratories or standards organisations. OCL-SOP addresses the issue of lack of standardisation in the representation of clinical laboratory SOP. The significant impact of OCL-SOP is the improvement in accuracy and completeness of the procedures presented in the SOPs. Using OCL-SOP in laboratory applications improves interoperability by enabling efficient exchange and interpretation of testing procedures and results between different healthcare settings. We created the OCL-SOP by reusing and re-engineering the ontology EXACT, which models experimental actions and their descriptors. Although clinical laboratory procedures apply the knowledge presented in the ontology EXACT, the ontology was missing some essential entities that are pertinent to clinical laboratory

SOPs. Therefore, we followed the principle of reusing ontologies recommended by the OBO foundry and re-purposed the ontology EXACT to address the needs of clinical laboratories. The result was the OCL-SOP which models the processes of experimental and data actions captured within SOPs and the description of how these processes occur, specifically what other entities the processes need to be successful. We modelled these entities as descriptors of the experimental actions, and it includes entities such as min-temperature, max-temperature, speed, equipment, and biochemical entities involved in a clinical laboratory process.

In order for developers to create smart clinical laboratory procedure applications that utilise SOPs, machines need to be able to read and understand the content of the SOPs. We have identified that SOPs are currently written in natural language, which makes it difficult for machines to process the content of the documents intelligibly. To remedy this problem, we took the first step by creating OCL-SOP, which provides a knowledge base of the SOPs in a machine-processable format. The second component of our proposed framework, the translation engine, provides a tool for converting the current free text SOPs into machine-readable formats without losing any essential information. We addressed objective 3 through the translation engine, which we presented in chapter 4. The translation engine is an NLP tool which extracts the experimental and data actions from the free text SOPs and all relevant information regarding these processes such as the material entities (biochemical entity and equipment) and other descriptor entities (e.g., the temperature, speed, and duration). The translation engine processes the free text SOPs based on the data model provided by OCL-SOP and generates a machine-readable output file containing all the extracted information.

The output files from the translation engine can have several uses as we have demonstrated in this thesis. One way is seen in the third component of Smart-SOP framework, the mobile application, which we also presented in chapter 4. The mobile application reads the output file from the translation engine and displays the content through a seamless graphical user interface. We addressed

objective 4 of this research through the mobile application as it gives the clinical laboratory scientists easy access to the SOPs while allowing managers to monitor usage of SOPs. We identified that one of the challenges the laboratory scientists are currently facing is that using the free text SOPs and finding the right information is very difficult. This inhibits adherence to the SOPs and inadvertently affects the quality of the laboratory procedures. The mobile application solves this problem by providing easy access to the laboratory SOPs with functionalities such as intuitive navigation, search functionality, and supplementary materials such as images and videos to demonstrate how a procedure is carried out. We also included a login feature which provides usage data and enable managers to monitor usage of the SOPs. The easy access and monitoring functionality support adherence to the SOPs, which we have identified as one problem clinical laboratories are currently facing. We provided an additional useful feature in the mobile application, which allows the lab scientists to record results of the procedures. The results are recorded in a standardised format and mapped to the data entity in OCL-SOP, which makes it easier to share across smart applications. The mobile application can also export the test results in .csv format to further support interoperability with other systems, including legacy systems.

In chapter 5, we discussed external projects that have used the SmartSOP framework or some components of the framework. We described the Maholo LabDroids and how it implements the most common laboratory experimental actions. We also described how we adapted the SmartSOP framework to process protocols from the Human Brain Project. We aligned OCL-SOP with NDDO to fully capture the ADNI and PPMI procedures described in the study manuals and the output of these procedures. We processed the study manuals with the translation engine to convert from natural language into machine-readable formats. We also discussed efforts to align OCL-SOP with an ontology for robot task representation from an ongoing project. The alignment of OCL-SOP with the robot task representation ontology provides a data model to support

automation of clinical laboratory procedures using robotics. We described a proposed implementation of a malaria microscopy robot based on this new data model.

We evaluated the SmartSOP framework using a user-centred evaluation approach, which we presented in chapter 6. Domain experts participated in this evaluation, where we experimented in three separate clinical laboratories, and the outcome of demonstrated the usefulness and usability of the framework is quite positive. The results of the evaluation show that the framework shows an accurate representation of the SOPs. The results also indicate that the framework can encourage and support compliance with the SOPs and good laboratory practices.

7.2 Research limitations

One of the contributions of this research is the OCL-SOP as a formal model for representing clinical laboratory SOPs. The domain of interest here is the biomedical domain, and we have identified that such a formal data model does not exist. To make OCL-SOP compatible with other biomedical ontologies, we reused some terms from existing OBO ontologies, including the entire ontology EXACT, which is also an OBO ontology. However, we have not thoroughly followed all the principles of the OBO Foundry (for example, the naming conventions).

One limitation of the OCL-SOP is that it does not formally represent the order of actions. The order of actions is currently determined from the SOP translator based on the arrangement of sentences and the order of action tokens in the sentences. This approach is not always accurate as seen in the example of "extract serum from centrifuged blood". Also, there is no way to determine if actions can be run concurrently or to show dependencies. OCL-SOP also does not differentiate high level actions like *inoculate* from low level actions like *pour* or *mix*.

There is also need for more work on the SOP translator to improve it's performance and efficiency. One of the issues is that the tool does not handle all forms of negations. The only way it can determine negation is if the word/phrase, "do not" or "don't" precedes the action. Although we implemented lemmatization in the NLP tool to handle inflectional words, there are still problems in correct identification both in terms of detecting non action words as actions and vice versa. There is also a the issue of identification of all verbs as actions (see section 4.3.4).

There are some limitations with the evaluation of the SOP translator. The current evaluation is lightweight on a small dataset and only measures the performance of the tool in identifying the action entities. The evaluation does not measure the identification of descriptors by the SOP translator or its individual components. Even though we used a standard measurement metrics for the evaluation, we could not compare the results with other popular NLP tools such as cTAKES and MetaMap. Although these two tools can also process the SOP text to identify descriptors of actions such as the biochemical entities and equipment, and high level actions, they are not designed to identify the low level clinical laboratory actions. For example, MetaMap can identify *centrifugation* as a laboratory procedure, but it does not recognise *pour* as a clinical laboratory action. This makes it challenging to make a direct comparison of the SOP translator to the other NLP tools.

7.3 Further work

We have identified directions for further work based on the research reported in this thesis.

7.3.1 Ontology

The biomedical domain will benefit from the addition of OCL-SOP to a recognised ontology library, specifically the BioPortal. Therefore, we recommend

further work on OCL-SOP to prepare it for hosting on the portal. Adopting all the OBO foundry principles and guidelines will ensure that OCL-SOP is eligible for hosting on the BioPortal.

We have also identified that more work is needed to create a version of OCL-SOP for laboratory task automation. The robot task representation standard is still in the development stage; thus, we recommend that once it is completed, OCL-SOP should be aligned to the new standard. This will ensure that all laboratory procedures are accurately modelled in the context of allowing robots to carry out the procedures. We can further develop the malaria microscopy robot use case we presented in chapter 5 and demonstrate how the robot can be re-purposed to carry out other laboratory procedures.

By aligning the OCL-SOP to the robot task standard, we can also address the limitations of lack of differentiation between low level and high level actions and formal representation of the order of actions. These two issues are being handled in the robot task standard. In addition, we can consider reusing the standard representations of OWL-S, which formalises the Process ontology that consists of the concept of atomic, simple and composite process as well as control constructs that define the order of processes (sequence, any-order, choice) [83].

7.3.2 NLP work

Further work is required on the SOP translator to improve its overall performance for NER. One approach we can use is word embeddings, which uses algorithms to learn about things via context and meaning of how they are used within text [137]. Because word embeddings uses the context of how words are used, we can use it to make the SOP translator understand how actions and their related descriptors are presented in the SOP text. This will enable the tool to correctly identify instances of synonyms, inflectional words, and words that can have two meanings, as a descriptor or action. When we use word embeddings, it will allow us to improve the OCL-SOP as we can see patterns that we were not able to identify through manual analysis of the SOP texts. For

example, word embeddings can learn the pattern of actions that appear in the same context, giving us better insight into how the actions are related in the different laboratory procedures.

We need to conduct further evaluation of the SOP translator. We will measure the performance of the individual components of the tool, the tokenizer, POS tagger, and lemmatization component. OpenNLP offers built in evaluation tools that can measure the accuracy of these components, which we can leverage. We also need to carry out more evaluation on the NER component to measure the performance of the tool in identifying the descriptors and matching them to the action entities. The result of the evaluation needs to be compared against the performance of similar NLP tools.

7.3.3 Framework

We also identified a further direction for the SmartSOP framework to support semi-automatic creation of new clinical laboratory SOPs. This can be achieved through an application that will allow lab specialist to create SOPs in machine-readable format. Currently, new SOPs need to be created by domain experts who do not have the advanced skills required to write the information in a machine-readable language. They will benefit from a software tool that will make it easy to use the representation in the OCL-SOP to create new SOP instantiations, which will also be in machine-readable format. To achieve this, we propose a web application that can be easily accessed without the need for installation. This proposed application can also apply automated reasoning to verify the validity of clinical laboratory procedures against standards defined in the OCL-SOP. The tool will give more visibility for this research work and will make the SmartSOP framework accessible to clinical laboratory practitioners where it will have the most impact.

Evaluation of SmartSOP Framework

Thank you for taking part in our research study.

Complete this questionnaire after testing the mobile app prototype. Please note that only the Malaria Microscopy Test is available for this evaluation.

Choose the answer that best suits your response to the following questions.

* Required

Part 1 - Participant Characterisation

1. **Age** *
Mark only one oval.
- 18 - 29 years old
- 30 - 39 years old
- 40 - 49 years old
- 50 years or older

2. **Do you own a smartphone?**
Mark only one oval.
- Yes
- No

3. **How long have you worked at a clinical laboratory?**
Mark only one oval.
- 0 - 5 years
- 5 - 10 years
- More than 10 years

4. **How often do you use SOPs in the laboratory in a week?** *
Mark only one oval.
- Never
- Once
- 2 - 3 times
- 4 - 5 times
- More than 5 times

Part 2 - Usability Testing

Choose from the scale the number that best describes your answer.
Strongly Disagree-1, Disagree-2, Neutral-3, Agree-4, and Strongly Agree-5

5. **The mobile app screen design is clear** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

6. **Navigating through the mobile app is easy.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

7. **Information about SOPs are easier to access on the mobile app than on the MS Word, PDF, or paper versions.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

8. **The information about test procedures are accurate.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

9. **The information about test procedures are complete (for example all values for volume of liquids or duration of procedures).** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

10. **The checklist shows all materials that are required to carry out the tests.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

11. **The correct terminologies are used.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

12. **The additional information such as videos are helpful.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

13. **The search functionality is useful** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

14. **Recording the test results on the mobile app is helpful.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

15. **While recording the test results, correcting your mistakes is easy** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

16. **Using the mobile app does not interfere with the actual testing procedure.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

17. **Having the SOPs on the mobile app will encourage me to use SOPs more frequently.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

18. **Any other comments or feedback?**

Thank you for completing the questionnaire.

Powered by Google Forms

Appendix A

Bibliography

[1] a suite of core nlp tools., 2016.

[2] Saminda Abeyruwan, Uma D Vempati, Hande Küçük-McGinty, Ubbo Visser, Amar Koleti, Ahsan Mir, Kunie Sakurai, Caty Chung, Joshua A Bittker, Paul A Clemons, et al. Evolving bioassay ontology (bao): modularization, integration and applications. In *Journal of biomedical semantics*, volume 5, page S5. BioMed Central, 2014.

[3] Hans Akkermans and Jaap Gordijn. Ontology engineering, scientific method and the research agenda. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 112–125. Springer, 2006.

[4] Alzheimers Disease Neuroimaging Initiative. Adni go procedures manual, 2019.

[5] Katrin Amunts, Alois C Knoll, Thomas Lippert, Cyriel MA Pennartz, Philippe Ryvlin, Alain Destexhe, Viktor K Jirsa, Egidio DAngelo, and Jan G Bjaalie. The human brain projectsynergy between neuroscience, computing, informatics, and brain-inspired technologies. *PLoS biology*, 17(7):e3000344, 2019.

[6] Grigoris Antoniou and Frank Van Harmelen. *A semantic web primer*. MIT press, 2004.

[7] Alan R Aronson. Effective mapping of biomedical text to the umls

metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association, 2001.

[8] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[9] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology*. Mit Press, 2015.

[10] Robert Arp, Barry Smith, and Andrew D Spear. Principles of best practice i: Domain ontology design. 2015.

[11] Julio César Arpírez, Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Webode in a nutshell. *AI magazine*, 24(3):37–37, 2003.

[12] Balsamiq Studios, LLC. Rapid, effective, and fun wireframing software. `https://balsamiq.com/`, 2019.

[13] Anita Bandrowski, Ryan Brinkman, Mathias Brochhausen, Matthew H Brush, Bill Bug, Marcus C Chibucos, Kevin Clancy, Mélanie Courtot, Dirk Derom, Michel Dumontier, et al. The ontology for biomedical investigations. *PloS one*, 11(4):e0154556, 2016.

[14] Jonathan Bard, Seung Y Rhee, and Michael Ashburner. An ontology for cell types. *Genome biology*, 6(2):R21, 2005.

[15] Jason M Baron and Anand S Dighe. Computerized provider order entry in the clinical laboratory. *Journal of pathology informatics*, 2, 2011.

[16] Randolph C Barrows Jr, James J Cimino, and Paul D Clayton. Mapping clinically useful terminology to a controlled medical vocabulary. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 211. American Medical Informatics Association, 1994.

[17] Richard Baskerville. What design science is not, 2008.

[18] JM Christian Bastien. Usability testing: some current practices and research questions. *International journal of medical informatics*, 2010.

[19] Matthias Becker and Britta Böckmann. Extraction of umls® concepts using apache ctakes for german language. *Studies in health technology and informatics*, 223:71, 2016.

[20] Jeffery L Belden, Rebecca Grayson, and Janey Barnes. Defining and testing emr usability: Principles and proposed methods of emr usability evaluation and rating. Technical report, Healthcare Information and Management Systems Society (HIMSS), 2009.

[21] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[22] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.

[23] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[24] Olivier Bodenreider and Anita Burgun. Biomedical ontologies. In *Medical Informatics*, pages 211–236. Springer, 2005.

[25] Pierangelo Bonini, Mario Plebani, Ferruccio Ceriotti, and Francesca Rubboli. Errors in laboratory medicine. *Clinical chemistry*, 48(5):691–698, 2002.

[26] Paul P. Bourbeau and Nathan A. Ledeboer. Automation in clinical microbiology. *Journal of Clinical Microbiology*, 51(6):1658–1665, 2013.

[27] David N Breslauer, Robi N Maamari, Neil A Switz, Wilbur A Lam, and Daniel A Fletcher. Mobile phone based clinical microscopy for global health applications. *PloS one*, 4(7):e6320, 2009.

[28] Ryan R Brinkman, Mélanie Courtot, Dirk Derom, Jennifer M Fostel, Yongqun He, Phillip Lord, James Malone, Helen Parkinson, Bjoern Peters, Philippe Rocca-Serra, et al. Modeling biomedical experimental processes with obi. In *Journal of biomedical semantics*, volume 1, page S7. BioMed Central, 2010.

[29] Manfred Broy. Can practitioners neglect theory and theoreticians neglect practice? *Computer*, 44(10):19–24, 2011.

[30] Kate Button, Robert W Van Deursen, Larisa Soldatova, and Irena Spasić. Trak ontology: defining standard care for the rehabilitation of knee conditions. *Journal of Biomedical Informatics*, 46(4):615–625, 2013.

[31] Wendy W Chapman, Marcelo Fiszman, John N Dowling, Brian E Chapman, and Thomas C Rindflesch. Identifying respiratory findings in emergency department reports for biosurveillance using metamap. *Medinfo*, 2004:487–491, 2004.

[32] B. J. Choi, S. M. Jin, S. H. Shin, J. C. Koo, S. M. Ryew, M. C. Kim, J. Kim, W. H. Son, K. T. Ahn, W. Chung, and H. R. Choi. Development of flexible laboratory automation platform using mobile agents in the clinical laboratory. In *2008 IEEE International Conference on Automation Science and Engineering*, pages 918–923, Aug 2008.

[33] Alex M Clark, Nadia K Litterman, Janice E Kranz, Peter Gund, Kellan Gregory, and Barry A Bunin. Bioassay templates for the semantic web. *PeerJ Computer Science*, 2:e61, 2016.

[34] Jordi Conesa, Xavier de Palol, and Antoni Olivé. Building conceptual schemas by refining general ontologies. In *International Conference on Database and Expert Systems Applications*, pages 693–702. Springer, 2003.

[35] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & knowledge engineering*, 46(1):41–64, 2003.

[36] Dina Demner-Fushman, Willie J Rogers, and Alan R Aronson. Metamap lite: an evaluation of a new java implementation of metamap. *Journal of the American Medical Informatics Association*, 24(4):841–844, 2017.

[37] Kevin Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.

[38] EMBL-EBI. Chemical Entities of Biological Interest (ChEBI). `https://www.ebi.ac.uk/chebi/`, 2019.

[39] Exadel, Inc. Professional app builder and mobile backend. `https://appery.io/`, 2019.

[40] Mariano Fernández-López and Asunción Gómez-Pérez. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17(2):129–156, 2002.

[41] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. In *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, 1997.

[42] James Matthew Fielding, Jonathan Simon, Werner Ceusters, and Barry Smith. Ontological theory for ontological engineering: Biomedical systems information integration. In *KR*, pages 114–120, 2004.

[43] Christian Fischer and Shirley Gregor. Forms of reasoning in the design science research process. In *International Conference on Design Science Research in Information Systems*, pages 17–31. Springer, 2011.

[44] Joseph E Fitzgibbon and Carole L Wallis. Laboratory challenges conducting international clinical research in resource-limited settings. *Journal of acquired immune deficiency syndromes (1999)*, 65(0 1):S36, 2014.

[45] Paul Fontelo, John Faustorilla, Alex Gavino, and Alvin Marcelo. Digital pathology–implementation challenges in low-resource countries. *Analytical Cellular Pathology*, 35(1):31–36, 2012.

[46] Arden W Forrey, Clement J Mcdonald, Georges DeMoor, Stanley M Huff, Dennis Leavelle, Diane Leland, Tom Fiers, Linda Charles, Brian Griffin, Frank Stalling, et al. Logical observation identifier names and codes (loinc) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical chemistry*, 42(1):81–90, 1996.

[47] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 166–181. Springer, 2002.

[48] Daniel Garijo. Widoco: a wizard for documenting ontologies. In *International Semantic Web Conference*, pages 94–102. Springer, Cham, 2017.

[49] Vijay Garla, Vincent Lo Re III, Zachariah Dorey-Stein, Farah Kidwai, Matthew Scotch, Julie Womack, Amy Justice, and Cynthia Brandt. The yale ctakes extensions for document classification: architecture and application. *Journal of the American Medical Informatics Association*, 18(5):614–620, 2011.

[50] Asunción Gómez-Pérez. Towards a framework to verify knowledge sharing technology. *Expert Systems with applications*, 11(4):519–529, 1996.

[51] Asunción Gómez-Pérez, Natalia Juristo, and Juan Pazos. Evaluation and assessment of knowledge sharing technology. *Towards very large knowledge bases*, pages 289–296, 1995.

[52] Pierre Grenon, Barry Smith, and Louis Goldberg. Biodynamic ontology: applying bfo in the biomedical domain. *Studies in health technology and informatics*, pages 20–38, 2004.

[53] Michael Grüninger and Mark S Fox. Methodology for the design and evaluation of ontologies. 1995.

[54] Michael Grüninger and Mark S Fox. The role of competency questions in enterprise engineering. In *BenchmarkingTheory and practice*, pages 22–31. Springer, 1995.

[55] Nicola Guarino and Christopher Welty. A formal ontology of properties. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 97–112. Springer, 2000.

[56] Nicola Guarino and Christopher A Welty. An overview of ontoclean. In *Handbook on ontologies*, pages 151–171. Springer, 2004.

[57] Josh Hanna, Chen Cheng, Alex Crow, Roger A Hall, Jie Liu, Tejaswini Pendurthi, Trent Schmidt, Steven F Jennings, Mathias Brochhausen, and William R Hogan. Simplifying mireot; a mireot protege plugin. In *International Semantic Web Conference (Posters & Demos)*, 2012.

[58] Robert Hawkins. Managing the pre-and post-analytical phases of the total testing process. *Annals of laboratory medicine*, 32(1):5–16, 2012.

[59] Heinrich Herre. Formal ontology and the foundation of knowledge organization. *KO KNOWLEDGE ORGANIZATION*, 40(5):332–339, 2014.

[60] Alan Hevner and Samir Chatterjee. Design science research in information systems. In *Design research in information systems*, pages 9–22. Springer, 2010.

[61] Hipposoft, LLC. Lab values medical reference, 2019.

[62] Robert Hoehndorf. What is an upper-level ontology? 2010.

[63] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible sroiq. *Kr*, 6:57–67, 2006.

[64] EA Howe, A De Souza, David L Lahr, S Chatwin, Philip Montgomery, BR Alexander, D-T Nguyen, Yasel Cruz, DA Stonich, G Walzer, et al. Bioassay research database (bard): chemical biology and probe-development enabled by structured metadata and result types. *Nucleic acids research*, 43(D1):D1163–D1170, 2014.

[65] Human Brain Project. Calls for expression of interest. `https://www.humanbrainproject.eu/en/collaborate/open-calls/`, 2017.

[66] Human Brain Project. Human brain project. `https://www.humanbrainproject.eu/en/`, 2019.

[67] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Nurfadhlina Mohd Sharef, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 6(16):2993–3000, 2013.

[68] Raymond F Jankowski. Implementing national guidelines at local level: Changes in clinicians' behaviour in primary care need to be reflected in secondary care, 2001.

[69] Paul Johannesson and Erik Perjons. *An introduction to design science*. Springer, 2014.

[70] Hyun-Young Kim, Hyeoun-Ae Park, Yul Ha Min, and Eunjoo Jeon. Development of an obesity management ontology based on the nursing process for the mobile-device domain. *Journal of medical Internet research*, 15(6):e130, 2013.

[71] Ross D King, Jem Rowland, Stephen G Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar

Pir, Larisa N Soldatova, et al. The automation of science. *Science*, 324(5923):85–89, 2009.

[72] Fabrice Kordon et al. An introduction to rapid system prototyping. *IEEE Transactions on Software Engineering*, 28(9):817–821, 2002.

[73] Ana Kostovska, Ilin Tolovski, Fatima Maikore, Larisa Soldatova, Panče Panov, Alzheimers Disease Neuroimaging Initiative, et al. Neurodegenerative disease data ontology. In *International Conference on Discovery Science*, pages 235–245. Springer, 2019.

[74] Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. A description logic primer. *arXiv preprint arXiv:1201.4089*, 2012.

[75] Johan Lauwereyns, Katsumi Watanabe, Brian Coe, and Okihide Hikosaka. A neural correlate of response bias in monkey caudate nucleus. *Nature*, 418(6896):413, 2002.

[76] Douglas B Lenat, Ramanathan V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd. Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49, 1990.

[77] Lucid Software Inc. Online diagram software and visual solution — lucidchart. `https://www.lucidchart.com/`, 2019.

[78] Eamonn Maguire, Alejandra González-Beltrán, Patricia L Whetzel, Susanna-Assunta Sansone, and Philippe Rocca-Serra. Ontomaton: a bioportal powered ontology widget for google spreadsheets. *Bioinformatics*, 29(4):525–527, 2012.

[79] Fatima S Maikore, Gantigmaa Selenge, Adebola Olayinka, Pamela Abbott, and Larisa N Soldatova. An ontology for clinical laboratory standard operating procedures. In *JOWO*, 2017.

[80] Fatima Sabiu Maikore, Emma Haddi, and Larisa Soldatova. A framework for it support of clinical laboratory standards. *International Journal*

of Privacy and Health Information Management (IJPHIM), 6(2):13–25, 2018.

[81] Ashutosh Malhotra, Erfan Younesi, Michaela Gündel, Bernd Müller, Michael T Heneka, and Martin Hofmann-Apitius. Ado: A disease ontology representing the domain knowledge specific to alzheimer's disease. *Alzheimer's & dementia*, 10(2):238–246, 2014.

[82] Salvatore T March and Gerald F Smith. Design and natural science research on information technology. *Decision support systems*, 15(4):251–266, 1995.

[83] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, et al. Owl-s: Semantic markup for web services. *W3C member submission*, 22(4), 2004.

[84] James Martin. *Rapid application development*. Macmillan Publishing Co., Inc., 1991.

[85] David W McClymont and Paul S Freemont. With all due respect to maholo, lab automation isn't anthropomorphic. *Nature biotechnology*, 35(4):312, 2017.

[86] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[87] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.

[88] Mark A Musen et al. The protégé project: a look back and a look forward. *AI matters*, 1(4):4, 2015.

[89] National Cancer Institute. Nci thesaurus. https://ncithesaurus.nci.nih.gov/, 2019.

[90] National Center for Biomedical Ontology. Ncbo bioportal. https://bioportal.bioontology.org/, 2019.

[91] JB Ndihokubwayo, F Kasolo, AA Yahaya, and J Mwenda. Strengthening public health laboratories in the who african region: a critical need for disease control. *African Health Monitor*, 12:47–52, 2010.

[92] NPC National Population Commission Nigeria and ICF. Nigeria demographic and health survey 2018 report. Technical report, National Population Commission Nigeria, 2019.

[93] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[94] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, Christopher G Chute, et al. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, 37(suppl_2):W170–W173, 2009.

[95] Jay F Nunamaker Jr, Nathan W Twyman, and Justin Scott Giboney. Breaking out of the design science box: High-value impact through multidisciplinary design science programs of research. 2013.

[96] Federal Ministry of Health Nigeria. National guidelines for diagnosis and treatment of malaria 3rd edition. Technical report, National Malaria and Vector Control Division, 2015.

[97] NMCP Federal Ministry of Health Nigeria. Malaria performance review, 2012. Technical report, National Malaria Control Programme, 2012.

[98] ISO/TC 159/SC 4 Ergonomics of human-system interaction (Subcommittee). *Ergonomic Requirements for Office Work with Visual Display*

Terminals (VDTs).: Guidance on Usability. International Organization for Standardization, 1998.

[99] World Health Organization et al. Guide for national public health laboratory networking to strengthen integrated disease surveillance and response (idsr). *Brazzaville, Republic of the Congo: World Health Organization Regional Office for Africa*, 2008.

[100] World Health Organization et al. Strategic framework for strengthening health laboratory services 2016–2020. Technical report, World Health Organization. Regional Office for the Eastern Mediterranean, 2017.

[101] John David Osborne, Binod Gyawali, and Thamar Solorio. Evaluation of ytex and metamap for clinical concept recognition. *arXiv preprint arXiv:1402.1668*, 2014.

[102] OWL Working Group and others. Owl 2 web ontology language document overview: W3c recommendation 27 october 2009. 2009.

[103] Seung Park, Anil Parwani, Mahadev Satyanarayanan, and Liron Pantanowitz. Handheld computing in pathology. *Journal of pathology informatics*, 3, 2012.

[104] Adam Pease, Ian Niles, and John Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Working notes of the AAAI-2002 workshop on ontologies and the semantic web*, volume 28, pages 7–10, 2002.

[105] Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. Comparing the performance of different nlp toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[106] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. Oops!(ontology pitfall scanner!): An on-line tool for ontology

evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34, 2014.

[107] Wanda Pratt and Meliha Yetisgen-Yildiz. A study of biomedical concept identification: Metamap vs. people. In *AMIA annual symposium proceedings*, volume 2003, page 529. American Medical Informatics Association, 2003.

[108] Public Health England. Uk standards for microbiology investigations, 2014.

[109] Ruth Reátegui and Sylvie Ratté. Comparison of metamap and ctakes for entity extraction in clinical notes. *BMC medical informatics and decision making*, 18(3):74, 2018.

[110] Thomas C Rindflesch and Alan R Aronson. Ambiguity resolution while mapping free text to the umls metathesaurus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 240. American Medical Informatics Association, 1994.

[111] Cornelius Rosse and Jose LV Mejino. The foundational model of anatomy ontology. In *Anatomy Ontologies for Bioinformatics*, pages 59–117. Springer, 2008.

[112] Manuel Salvadores, Paul R Alexander, Mark A Musen, and Natalya F Noy. Bioportal as a dataset of linked biomedical ontologies and terminologies in rdf. *Semantic web*, 4(3):277–284, 2013.

[113] Rishi Kanth Saripalle. Current status of ontologies in biomedical and clinical informatics. *University of Connecticut*, 2004.

[114] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architec-

ture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[115] Richard H Scheuermann, Werner Ceusters, and Barry Smith. Toward an ontological treatment of disease and diagnosis. *Summit on translational bioinformatics*, 2009:116, 2009.

[116] Peri L Schuyler, William T Hole, Mark S Tuttle, and David D Sherertz. The umls metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, 81(2):217, 1993.

[117] Andrew Sears and Julie A Jacko. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC press, 2007.

[118] Andrew Sears and Julie A Jacko. *Human-computer interaction: Development process*. CRC Press, 2009.

[119] Herbert A Simon. *The sciences of the artificial*. MIT press, 2019.

[120] Aarti Singh and Poonam Anand. State of art in ontology development tools. *International Journal*, 2(7), 2013.

[121] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251, 2007.

[122] Barry Smith, Anand Kumar, and Thomas Bittner. Basic formal ontology for bioinformatics. 2005.

[123] Barry Smith, Tatiana Malyuta, Ron Rudnicki, William Mandrick, David Salmen, Peter Morosoff, Danielle K Duff, James Schoening, and Kesny Parent. Iao-intel: an ontology of information artifacts in the intelligence domain. 2013.

[124] Larisa N Soldatova, Wayne Aubrey, Ross D King, and Amanda Clare. The exact description of biomedical protocols. *Bioinformatics*, 24(13):i295–i303, 2008.

[125] Larisa N Soldatova, Daniel Nadis, Ross D King, Piyali S Basu, Emma Haddi, Véronique Baumlé, Nigel J Saunders, Wolfgang Marwan, and Brian B Rudkin. Exact2: the semantics of biomedical protocols. *BMC bioinformatics*, 15(14):S5, 2014.

[126] Irena Spasić, Bo Zhao, Christopher B Jones, and Kate Button. Kneetex: an ontology–driven system for information extraction from mri reports. *Journal of biomedical semantics*, 6(1):34, 2015.

[127] Suresh Srinivasan, Thomas C Rindflesch, William T Hole, Alan R Aronson, and James G Mork. Finding umls metathesaurus concepts in medline. In *Proceedings of the AMIA Symposium*, page 727. American Medical Informatics Association, 2002.

[128] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[129] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2012.

[130] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernandez-Lopez. The neon methodology framework: A scenario-based methodology for ontology development. *Applied ontology*, 10(2):107–145, 2015.

[131] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Boris Villazón-Terrazas. How to write and use the ontology requirements specification document. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 966–982. Springer, 2009.

[132] Jillian R Tate, Roger Johnson, Julian Barth, and Mauro Panteghini. Harmonization of laboratory testingcurrent achievements and future strategies. *Clinica Chimica Acta*, 432:4–7, 2014.

[133] The Apache Software Foundation. Apache OpenNLP. `https://opennlp.apache.org/`, 2019.

[134] U.S. National Library of Medicine. Pubchem. `https://pubchem.ncbi.nlm.nih.gov/`, 2019.

[135] Michael Uschold and Martin King. *Towards a methodology for building ontologies*. Citeseer, 1995.

[136] C Lee Ventola. Mobile devices and apps for health care professionals: uses and benefits. *Pharmacy and Therapeutics*, 39(5):356, 2014.

[137] Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, and Hongwei Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174:806–814, 2016.

[138] Patricia L Whetzel, Natalya F Noy, Nigam H Shah, Paul R Alexander, Csongor Nyulas, Tania Tudorache, and Mark A Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl_2):W541–W545, 2011.

[139] W John Wilbur, George F Hazard Jr, Guy Divita, James G Mork, Alan R Aronson, and Allen C Browne. Analysis of biomedical text for chemical names: a comparison of three methods. In *Proceedings of the AMIA Symposium*, page 176. American Medical Informatics Association, 1999.

[140] Jeremy Woodhill. Is design science a methodology, method, paradigm or approach?, 2012.

[141] Steven H Woolf, Richard Grol, Allen Hutchinson, Martin Eccles, and Jeremy Grimshaw. Potential benefits, limitations, and harms of clinical guidelines. *Bmj*, 318(7182):527–530, 1999.

[142] Zuoshuang Xiang, Mélanie Courtot, Ryan R Brinkman, Alan Ruttenberg, and Yongqun He. Ontofox: web-based support for ontology reuse. *BMC research notes*, 3(1):175, 2010.

[143] Nozomu Yachie, Koichi Takahashi, Toshiaki Katayama, Takeshi Sakurada, Genki N Kanda, Eiji Takagi, Takako Hirose, Tatsuo Katsura, Tetsuo Moriya, Hiroaki Kitano, et al. Robotic crowd biology with maholo labdroids. *Nature biotechnology*, 35(4):310, 2017.

[144] Li Zhou, Joseph M Plasek, Lisa M Mahoney, Frank Y Chang, Dana DiMaggio, and Roberto A Rocha. Mapping partners master drug dictionary to rxnorm using an nlp-based approach. *Journal of biomedical informatics*, 45(4):626–633, 2012.

[145] Li Zhou, Joseph M Plasek, Lisa M Mahoney, Neelima Karipineni, Frank Chang, Xuemin Yan, Fenny Chang, Dana Dimaggio, Debora S Goldman, and Roberto A Rocha. Using medical text extraction, reasoning and mapping system (mterms) to process medication information in outpatient clinical notes. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1639. American Medical Informatics Association, 2011.